



普通高等学校计算机教育课程“十二五”规划教材·精品系列
江西省精品课程主讲教材

数据结构

肖守柏 熊 蕾 吴金舟 主 编
胡剑锋 主 审

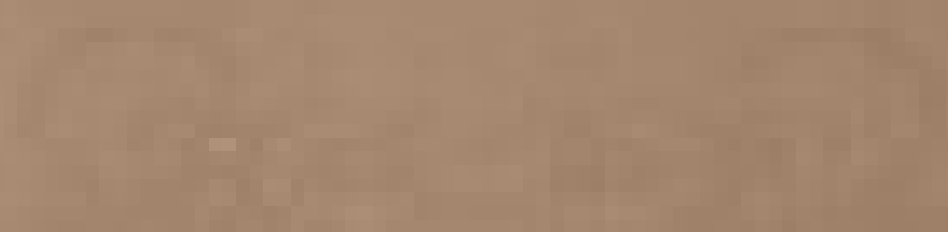
中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



清华大学出版社
Tsinghua University Press

数据结构

（第2版）
清华大学出版社



清华大学出版社

www.tsp.tsinghua.edu.cn



普通高等学校计算机教育课程“十二五”规划教材·精品系列

江西省精品课程主讲教材

数 据 结 构

肖守柏 熊 蕾 吴金舟 主 编

胡剑锋 主 审

内 容 简 介

“数据结构”不仅是计算机专业重要的专业基础课,也是从事计算机软件开发必备的专业知识。全书共分 12 章,主要介绍了数据结构的基本概念,线性表、栈、串、队列和数组,树结构、图结构,以及查找和排序等基本运算。每个章节基本上是从实例入手,然后系统地介绍本实例所涉及的知识点。本书注重应用性、由浅入深、逻辑性强,每章后均配有小结和思考与练习。

本书适合作为高等学校计算机专业学生的教材,也可供广大从事计算机软件工作的科技人员自学参考。

图书在版编目(CIP)数据

数据结构 / 肖守柏,熊蕾,吴金舟主编. — 北京:
中国铁道出版社,2012.8

普通高等学校计算机教育课程“十二五”规划教材.
精品系列

ISBN 978-7-113-14922-2

I. ①数… II. ①肖… ②熊… ③吴… III. ①数据结
构—高等学校—教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字(2012)第 186086 号

书 名: 数据结构

作 者: 肖守柏 熊 蕾 吴金舟 主编

策 划: 郑 涛

读者热线: 400-668-0820

责任编辑: 孟 欣 彭立辉

封面设计: 淡晓库

封面制作: 刘 颖

责任印制: 李 佳

出版发行: 中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: <http://www.51eds.com>

印 刷: 三河市华丰印刷厂

版 次: 2012 年 8 月第 1 版 2012 年 8 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 19.25 字数: 457 千

印 数: 1~3 000 册

书 号: ISBN 978-7-113-14922-2

定 价: 36.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 63549504

计算机科学技术以惊人的速度迅猛发展,它的应用范围已渗入到社会和生活的各个领域。相应地,数据处理的对象也从简单的数值发展到字符、表格和图形等带有结构的数据。在这里要解决的关键问题是:针对每一种新的应用领域的处理对象,如何选择合适的数据表示(结构),如何有效地组织数据、处理数据。数据结构就是研究数据以及数据之间关系的一门学科,主要研究数据之间的逻辑结构及其基本操作在计算机中的表示和实现方法。数据结构课程不仅是计算机专业重要的专业基础课,也是从事计算机软件开发所必备的专业知识。

本书主要面向应用型高等学校计算机类专业的学生,培养技术应用型人才;内容的构造力求体现“以应用为主体”,强调理论知识的理解和运用,实现本、专科教学以实践体系为主和以技术应用能力培养为主的目标。

案例教学是计算机课程教学最有效的方法之一,好的案例对学生理解知识、掌握如何应用知识都十分重要。本书围绕教学内容组织案例,对学生的知识和能力训练具有较强的针对性。全书共分12章,将2~12章又分成IV部分:第I部分(第2~6章),介绍基本的线性结构及有关的典型应用;第II部分(第7、8章),讲述具有广泛应用价值的树形结构,这两部分占据了本书的主要篇幅;第III部分(第9、10章),介绍复杂的数据结构,比如图、稀疏矩阵及广义表等;第IV部分(第11、12章),介绍有关外存储器中的数据结构和文件组织。此外,书中还有4个附录,分别是:数据结构实训指导、教学实验报告参考格式、课程应用与学习实验、“数据结构”模拟试题。书中的所有程序都在Turbo C 2.0环境下调试通过。

数据结构是实践性很强的课程,本书注重理论与实践相结合,每章都给出了不同层次、不同难度的思考题并且给出参考答案。通过习题与实训,使学生掌握所学知识,并能灵活运用所学知识解决实际问题。

本书带*的章节为专科层次的选讲部分,必讲部分建议授课64学时,其中上机实训安排20学时以上。教师可根据学时数、专业和学生的实际情况选讲带*号部分中一些较难的例子。

本书适合作为高等学校计算机专业学生的教材,也可供广大从事计算机软件工作的科技人员自学参考。

本书由江西科技学院肖守柏、熊蕾、吴金舟担任主编,由中国科学院博士胡剑锋教授担任主审。在本书编写过程中,得到南昌大学信息工程学院计算机科学与技术系姚立文教授的大力帮助,在此表示深深的谢意。本书是江西省精品课程“数据结构与算法”的主讲教材,是作者多年教学改革成果和结晶,教学资源可在<http://jpkc.jxbsu.com:8080/kc/datastructure/index.php>网站下载。

由于时间仓促,编者水平有限,书中错误和不妥之处在所难免,敬请广大同行和读者批评指正。

编 者

2012年8月

第 1 章 绪论	1
1.1 数据结构概述	1
1.1.1 学习数据结构的必要性	1
1.1.2 数据结构的基本概念和术语	2
1.1.3 数据类型与抽象数据类型	5
1.2 "HELLO,WORLD!"程序实例	7
1.2.1 C 语言的编写风格	7
1.2.2 C 语言预备知识	10
1.3 “数组元素排序”实例	12
1.3.1 算法的特性	14
1.3.2 算法的评价标准	14
1.3.3 算法度量及分析	15
小结	18
思考与练习	19
I 线性结构	
第 2 章 线性表	22
2.1 “银行排队”顺序存储实例	22
2.1.1 线性表的定义	23
2.1.2 线性表的基本操作	24
2.1.3 顺序表	24
2.2 “学生健康登记”链式存储实例	27
2.2.1 单链表的定义	28
2.2.2 单链表的基本操作	29
2.3 其他链表	32
2.3.1 循环链表	32
2.3.2 双向链表	33
小结	35
思考与练习	35
第 3 章 栈和队列	38
3.1 “回文”实例	38
3.1.1 栈的定义及基本运算	39
3.1.2 栈的存储实现和运算实现	40
*3.1.3 栈与递归	47

3.2 “杨辉三角形”实例	49
3.2.1 队列的定义及基本运算	51
3.2.2 队列的存储实现和运算实现	51
小结	58
思考与练习	58
第4章 串	60
4.1 串的基本概念	60
4.1.1 串的定义和术语	60
4.1.2 串的基本运算	61
4.2 “文本加密”实例	62
4.2.1 串的顺序存储	64
4.2.2 顺序串的基本运算	65
4.2.3 串的链式存储	69
*4.2.4 模式匹配	69
小结	73
思考与练习	73
第5章 内部排序	75
5.1 排序的基本概念	75
5.2 “学生成绩插入排序”实例	76
5.2.1 直接插入排序	77
5.2.2 希尔排序	79
5.3 “学生成绩交换排序”实例	81
5.3.1 冒泡排序	83
5.3.2 快速排序	85
5.4 “学生成绩选择排序”实例	87
5.4.1 直接选择排序	88
5.4.2 堆排序	90
5.5 其他排序介绍	95
5.5.1 有序序列的合并	95
5.5.2 二路归并排序	96
*5.5.3 基数排序	98
5.6 各种排序的比较	99
小结	101
思考与练习	102
第6章 查找	104
6.1 “学生成绩不及格的查找”实例	104
6.1.1 顺序查找	106
6.1.2 折半查找	107

6.2 “学生成绩及格的查找”实例	109
6.2.1 索引查找的概念	110
6.2.2 分块查找	111
6.3 “学生成绩优秀的查找”实例	112
6.3.1 哈希表与哈希方法	113
6.3.2 哈希函数的构造方法	115
*6.3.3 处理冲突的方法	117
6.3.4 哈希表的查找分析	120
小结	120
思考与练习	121

II 树形结构

第7章 二叉树	124
7.1 “高校篮球比赛”实例	124
7.1.1 二叉树的基本概念	126
7.1.2 二叉树的主要性质	126
7.1.3 二叉树的存储	128
7.2 “高校篮球总决赛”实例	130
7.2.1 遍历二叉树	131
7.2.2 线索二叉树的定义及结构	133
7.2.3 线索二叉树的基本操作	133
7.3 “学生成绩及格的查找”实例	135
7.3.1 二叉排序树	136
*7.3.2 平衡二叉树	139
7.4 “报文”实例	144
7.4.1 哈夫曼树的基本概念	146
7.4.2 哈夫曼编码	147
小结	147
思考与练习	148
第8章 树	150
8.1 “高校教师讲课比赛”实例一	150
8.1.1 树的定义及相关术语	152
8.1.2 树的表示	153
8.1.3 树的基本操作	154
8.1.4 树的存储结构	154
8.2 “高校教师讲课比赛”实例二	156
8.2.1 树转换为二叉树	158
8.2.2 树的遍历	159
8.2.3 森林	159

*8.3 树表动态查找.....	161
8.3.1 B-树和 B+树	161
8.3.2 键树	167
小结	169
思考与练习	169

III 复杂结构

第 9 章 图	172
9.1 “城际铁路”实例	172
9.1.1 图的定义和术语	173
9.1.2 图的存储表示	176
9.2 “游乐园路线”实例	180
9.2.1 图的遍历	181
9.2.2 最小生成树	184
9.2.3 最短路径	189
*9.3 有向无环图的应用	191
9.3.1 拓扑排序	191
9.3.2 关键路径	193
小结	196
思考与练习	196
第 10 章 数组、矩阵和广义表	199
10.1 “学生考勤的全勤天数”实例	199
10.1.1 数组的逻辑结构	199
10.1.2 数组的内存映像	201
10.2 “学生考勤的放假天数”实例	202
10.2.1 对称矩阵	203
10.2.2 三角矩阵	203
10.3 “学生考勤的请假天数”实例	204
10.3.1 稀疏矩阵的三元组表存储	205
*10.3.2 稀疏矩阵的十字链表存储	208
10.3.3 广义表的定义	212
*10.3.4 广义表的存储	213
10.3.5 广义表的运算	214
小结	215
思考与练习	216

IV 文件结构

第 11 章 文件	220
11.1 文件的基本概念	220

11.1.1 文件及其类别	220
11.1.2 文件记录的逻辑结构和物理结构	221
11.1.3 文件组织与操作	221
11.2 顺序文件	222
11.3 散列文件	223
*11.4 索引文件	224
11.4.1 ISAM 文件	225
11.4.2 VSAM 文件	226
*11.5 多关键字文件	227
11.5.1 多重表文件	228
11.5.2 倒排文件	228
小结	228
思考与练习	229
*第 12 章 外部排序	231
12.1 外部排序的基本思想	231
12.2 外部排序的方法	232
小结	236
思考与练习	236
附录 A “数据结构”实训指导	238
附录 B 教学实验报告参考格式	251
附录 C 课程应用与学习实验	254
附录 D “数据结构”模拟试题	262
各章思考与练习参考答案	270
参考文献	296

第1章 绪 论

【内容提要及目标】

本章首先介绍了数据结构中的几个基本概念和术语，然后以一个简单的 C 语言程序入手，复习了 C 语言的基本内容，进而对程序的算法特性、评价标准和时间复杂度分别进行介绍。通过本章学习，读者应该掌握以下内容：

- 了解数据结构中常用的基本概念和术语。
- 掌握抽象数据类型的定义、表示和实现方法。
- 熟悉 C 语言的书写规范。
- 理解算法 5 个要素的确切含义。
- 掌握计算语句频度和估算算法时间复杂度的方法。

【本章重点及难点】

理解数据结构的逻辑结构、存储结构和数据的运算三方面的概念及相互的关系；算法时间复杂度的分析。

1.1 数据结构概述

程序设计是计算机学科各个领域的基础。在计算机发展的早期，程序设计所处理的数据都是整型、实型等简单数据，绝大多数的应用软件都是用于数值计算。随着信息技术的发展，计算机逐渐进入到金融、商业、管理、通信以及制造业等各个行业，广泛地应用于数据处理和过程控制，计算机加工处理的对象也由纯粹的数值型数据发展到字符、表格和图像等各种具有一定结构的数据，这就给程序设计带来一些新的问题。随之，数据结构的概念在这种背景下产生了。

1.1.1 学习数据结构的必要性

数据结构不仅是计算机专业教学计划中的核心课程之一，也是其他非计算机专业的主要选修课之一。数据结构的研究不仅涉及计算机硬件的研究范围，而且还与计算机软件的研究有着密切的联系，无论在编程还是在操作系统，它都涉及数据元素在存储器中的分配问题，以及如何组织数据，使查找和存取数据元素更为方便。学习数据结构这门课程的目的有 3 个，第一是讲授常用的数据结构，这些数据结构形成了程序员基本数据结构工具箱 (Toolkit)。对于许多常见的问题，工具箱里的数据结构是理想的选择。例如，.NET Framework 中 Windows 应用程序开发中的工具箱，程序员可以直接拿来或经过少许的修改就可以使用，非常方便。第二是讲授常用的算法，算法和数据结构一样，是人们在长期实践过程中的总结，程序员可以直接拿来或经过少许的修改就可以使用。我们可以通过算法训练来提高程序设计水平。第三是通过程序设计的技能训练来促进程序员综合能力的提高。

1.1.2 数据结构的基本概念和术语

1. 数据结构实例

为了使大家对数据结构有一个感性的认识，下面先举以下几个例子来说明什么是数据结构。

【例 1.1】高校教师信息管理，如表 1-1 所示。

表 1-1 教师信息表

工号	姓名	性别	授课课程	课程编号	住 址	电 话
10001	匡青青	女	数据结构	2142	北京西路 411 号	86262xxx
10002	肖子皓	男	网络操作系统	2335	南京东路 999 号	85966xxx
10003	谭辉明	男	C 语言程序设计	0342	西安路 688 号	88138xxx
10004	彭明明	男	软件工程	2333	上海路 168 号	88138xxx
10005	章小花	女	网络技术	2141	八一大道 21 号	88174xxx
...	...	...	...	...	...	...

在表 1-1 中，一行代表一位教师的信息，每位教师的信息由工号、姓名、性别、授课课程、课程编号、住址、电话等组成，一列为一个特征，整个二维表按教师的工号顺序排列，每个教师的信息依据工号的大小存在着前后关系，即工号较小者在前，工号较大者在后，形成线性关系，这是一种典型的数据结构，这种关系可称为线性数据结构。有了模型以后，就可以围绕该模型设计算法，即实现教师信息的添加、修改、删除、检索等操作。

【例 1.2】某高校专业设置情况，如图 1-1 所示。

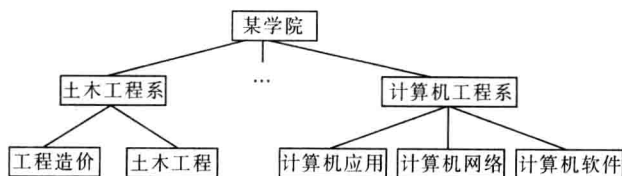


图 1-1 高校专业设置

在图 1-1 中，可以把“某学院”看成是树根，把下设的若干个系看成是它的树枝中间结点，把每个系的若干专业方向看成是树叶，这就形成一个树形结构。树形结构通常用来表示结点的分层组织，结点之间是一对多的关系，它也是一个典型的数据结构。

【例 1.3】有 7 个城市建立通信网络，用顶点表示城市，边上的权值表示两个城市之间建立通信线路所需花的费用，如图 1-2 所示。

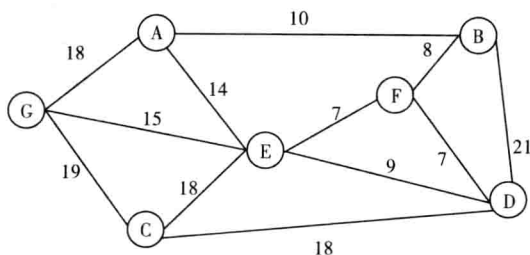


图 1-2 城市间通信网络

在此结构中，数据之间呈现多对多的非线性关系，这也是常用的一种数据结构，通常将其称为图形结构。

2. 基本概念和术语

上述 3 个例子都是数据结构的具体实例，那么，数据结构的定义是什么？下面介绍数据结构的基本概念和术语。

(1) 数据

数据 (Data) 是对信息的一种符号表示，是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。通俗地说，凡是能被计算机识别、存储和加工处理的符号，如字符、图形、图像、声音、视频信号等一切信息都可以称为数据。数据是计算机程序加工的“原料”。例如，一个利用数值分析方法解代数方程的程序，其处理的数据是整数和实数；一个文字处理器 (例如 Office 办公软件中的 Word) 处理的数据是字符串。

(2) 数据元素

数据元素 (Data Element) 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。数据元素有时也被称为元素、结点、顶点、记录等。一个数据元素可由若干个数据项组成，数据项是数据不可分割的最小单位。例如，在表 1-1 中，把一行作为一个数据元素来看待，那么工号、姓名、性别、授课课程、课程编号、住址、电话等就称为数据项 (Data Item)，有时也称为字段 (Field) 或域 (Domain)。

(3) 数据对象

数据对象 (Data Object) 是性质相同的数据元素的集合，是数据的一个子集。例如，整数数据对象的集合可表示为 $N=\{\cdots, -2, -1, 0, 1, 2, \cdots\}$ ，字母字符数据对象的集合可表示为 $C=\{'a', 'b', 'c', \cdots\}$ 。

3. 数据结构

数据结构 (Data Structure) 是指数据元素之间存在着的一种或多种特定关系的集合，这是对数据结构一种简单的定义。具体来说，数据结构是按某种逻辑关系组织起来的一批数据 (或称带结构的数据元素的集合)，它们应用计算机语言并按一定的表示方式存储在计算机的存储器中。数据结构包括数据的逻辑结构、数据的存储结构、数据的运算 3 个方面的内容，如图 1-3 所示。

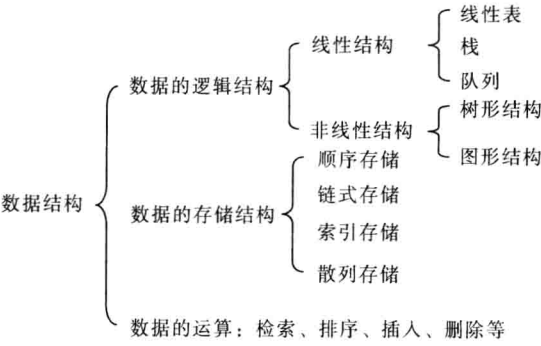


图 1-3 数据结构的内容

(1) 数据的逻辑结构

逻辑关系是指数据元素之间的关联方式，数据元素之间逻辑关系的整体称为逻辑结构，数据的逻辑结构就是数据的组织形式。从逻辑上划分，数据结构分为线性结构和非线性结构。

① 线性结构：数据元素之间为一对一的线性关系，第一个元素无直接前驱，最后一个元素无直接后继，其余元素只有唯一的一个前驱和唯一的一个后继。

② 非线性结构：数据元素之间为一对多或多对多的非线性关系，每个数据元素有多个直接前驱或者多个直接后继。

此外，又可根据数据元素之间关系的不同特性，将数据结构划分为 4 种类型，如图 1-4 所示。

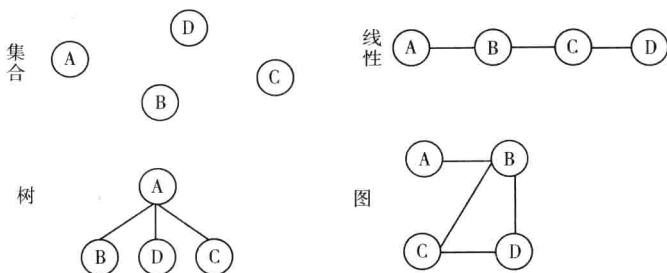


图 1-4 4 种基本数据结构

① 集合 (Set)：结构中的数据元素除了存在“同属于一个集合”的关系外，不存在任何其他关系。

② 线性结构 (Linear Structure)：结构中的数据元素之间存在一对一的关系，即有且仅有一个开始和一个终端结点，并且所有结点都最多只有一个直接前驱和一个后继。例如，线性表、栈、队列都属于线性结构。

③ 树形结构 (Tree Structure)：结构中数据元素之间存在一对多的关系。

④ 图状结构 (Graphic Structure)：结构中数据元素之间存在多对多的关系。

由于集合中元素的关系极为松散，可用其他数据结构来表示，数据结构的形式化定义为：

数据结构 (Data Structure) 简记为 DS，是一个二元组 $DS = (D, R)$ 。其中， D 是数据元素的有限集合， R 是数据元素之间关系的有限集合。

(2) 数据的存储结构

数据的逻辑结构是不能描述计算机如何操作数据的。研究数据结构的目的是为了在计算机中实现对它的操作，因此必须研究如何在计算机中表示数据结构，即数据的存储结构 (又称物理结构)，它包括数据元素的表示和关系的表示。数据的逻辑结构可以通过映像得到与它对应的存储结构，数据元素在计算机中的映像是元素 (Element) 或结点 (Node)，数据项的映像称为数据域 (Data Field)。数据元素间关系的表示是存储结构的主要部分，它由存储结点之间的关联方式表达。存储结点之间有以下 4 种关联方式：

① 顺序存储：每个存储结点只含一个数据元素，所有存储结点相继存放在一个连续的存储区里，逻辑上相邻的数据元素存放到计算机内存仍然相邻。借助数据元素在存储器中的相对位置来表示数据元素之间的逻辑关系，用这种方式表示逻辑关系的存储结构称为顺序存储结构。

② 链式存储：每个存储结点不仅含有一个数据元素，还包含一组指针，每个指针指向一个与本结点有逻辑关系的结点。该方法不要求逻辑上相邻的数据元素在物理位置上相邻，数

据元素之间的逻辑关系是由附加的指针表示的。

③ 索引存储：每个存储结点只包含一个数据元素，所有存储结点连续存放，此外使用这种方式存放元素的同时，还需建立附加的索引表，索引表中的每一项称为索引项。索引项的一般形式是：(关键字、索引)，其中的关键字是能唯一标识一个结点的数据项，索引指示各存储结点的存储位置或位置区间端点。

④ 散列存储：每个存储结点只包含一个数据元素，各个结点均匀分布在存储区里。该方法的基本思想是通过构造散列函数，根据结点的关键字直接计算出该数据元素的存储地址。

上述4种基本的存储方法，既可以单独使用，也可以组合起来对数据结构进行存储映像。同一种逻辑结构采用不同的存储方法，可以得到不同的存储结构。选择何种存储结构来表示相应的逻辑结构，视具体要求而定，主要考虑运算方便及算法的时空要求。

如何描述存储结构？虽然存储结构涉及数据元素及其关系在内存中的物理位置，但由于是在高级程序语言的层次上讨论数据结构的操作，因此，不直接以内存地址来描述存储结构，而借用高级语言中提供的“数据类型”来描述它。例如，顺序存储结构用数组来描述，链式存储结构利用指针来描述。在实际程序设计过程中，针对一种数据的逻辑结构，所选择的存储结构会影响到具体算法的设计。也就是说，在设计具体算法之前，必须先确定存储结构。在C语言中，一般使用 typedef 语句来为存储结构定义新的数据类型名字。

数据的逻辑结构和物理结构是密切相关的两个方面，任何一个算法的设计取决于选定的数据（逻辑）结构，而算法的实现依赖于所采用的存储结构。

（3）数据的运算

数据的运算是针对数据施加的操作。运算的定义取决于逻辑结构，运算的实现必依赖于存储结构。由于运算只描述处理功能，不包括处理步骤和方法，因此运算实现的核心是处理步骤的规定，即算法的设计。

1.1.3 数据类型与抽象数据类型

1. 数据类型

数据类型（Data Type）是和数据结构密切相关的一个概念，几乎所有高级语言都提供这一概念。数据类型是一个值的集合，以及在这个集合上定义的一组操作的总称。例如，C语言中的整型变量，其值集为某个区间上的整数（区间大小依赖于不同的机器），定义在其上的操作为：加、减、乘、除和取模等运算。

按“值”是否可分解，可以把数据类型分为两类：

① 原子类型：其值不可分解，比如C语言的基本类型（如整型、字符型、实型）、指针类型和空类型。

② 结构类型：其值可分解成若干成分（或称分量），比如C语言的数组类型、结构类型等。结构类型的成分可以是原子类型，也可以是某种结构类型，我们可以把数据类型看做程序设计语言已实现的数据结构。

引入数据类型的目的，从硬件角度考虑，是作为解释计算机内存中信息含义的一种手段；对用户来说，实现了信息的隐蔽，即将一切用户不必了解的细节都封装在类型中。例如，用户在使用整数类型时，既不需要了解整数在计算机内如何表示，也不必了解其操作（如两个整数

相加)在硬件中是如何实现的。

2. 抽象数据类型

抽象数据类型 (Abstract Data Type, ADT) 是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义取决于它的一组逻辑特性,而与其在计算机内部如何表示和实现无关。即不论其内部结构如何变化,只要它的数学特性不变,都不影响其外部的使用。

抽象数据类型和数据类型实质上是一个概念。例如,整数类型是一个 ADT,其数据对象是指能容纳的整数,基本操作有加、减、乘、除和取模等。尽管它们在不同处理器上的实现方法可以不同,但其定义的数学特性相同,在用户看来都是相同的。因此,“抽象”的意义在于数据类型的数学抽象特性。

但在另一方面,抽象数据类型的范畴更广,它不再局限于前述各处理器中已定义并实现的数据类型,还包括用户在设计软件系统时自己定义的数据类型。为了提高软件的重用性,在现在的程序设计方法学中,要求在构成软件系统的每个相对独立的模块上,定义一组数据和施于这些数据上的一组操作,并在模块的内部给出这些数据的表示及其操作的细节,而在模块的外部使用的只是抽象的数据及抽象的操作。这就是面向对象的程序设计方法。

抽象数据类型的定义可以由一种数据结构和定义在其上的一组操作组成,而数据结构又包括数据元素间的关系,因此抽象数据类型一般可以由元素、关系及操作 3 种要素来定义。

3. 抽象数据类型的表示

本书按以下格式表示抽象数据类型:

ADT 抽象数据类型名

数据元素集合:

数据元素集合的定义

基本操作:

基本操作的定义

其中,数据元素用自然语言描述,基本操作用伪码描述,并规定基本操作的格式为:

中文名(操作名): 含义

【例 1.4】抽象数据类型“字符串”的定义。

ADT String

数据元素集合:

字符的一个有限序列。

基本操作:

求串长 (StrLen): 求取字符串中字符的个数。

求子串 (SubStr): 获取字符串中的一个连续字符序列。

求位串 (Index): 定位子串在主串中第一次出现的位置。

求连接 (Concat): 连接两个字符串形成一个新串。

求比较 (StrCmp): 比较两个串的大小。

求空串 (StrEmpty): 判断所给字符串是否为空串。

求替换 (StrReplace): 替换字符串中指定的所有子串。

1.2 "HELLO,WORLD!"程序实例

【实例目的】

了解 C 语言上机环境，掌握 Turbo C 系统中各菜单的使用方法；熟悉 C 语言的编写风格及 C 语言程序基本框架。

【实例内容】

编写程序，输出“HELLO,WORLD!”。

【实例步骤】

参考附录 A，双击桌面上的 Turbo C 快捷图标，输入“HELLO,WORLD!”的 C 语言应用程序代码。为了使读者更清楚地了解程序的执行过程，下面给出编辑、编译和运行的具体步骤。

① 编辑程序。在 Turbo C 2.0 编译界面中按 Alt+F 组合键，选择 New 命令，然后输入源程序，如图 1-5 所示。

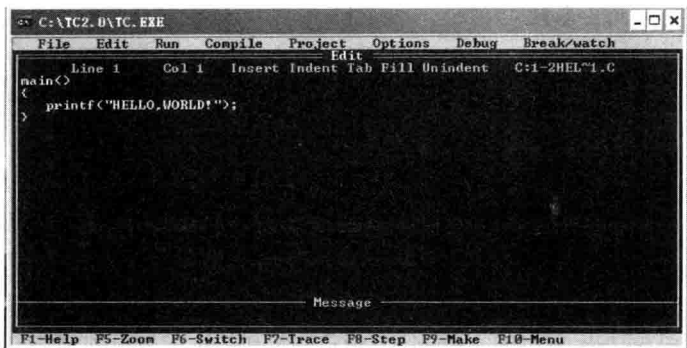


图 1-5 源代码界面

② 编译和运行程序，运行结果如图 1-6 所示。最后保存好程序。

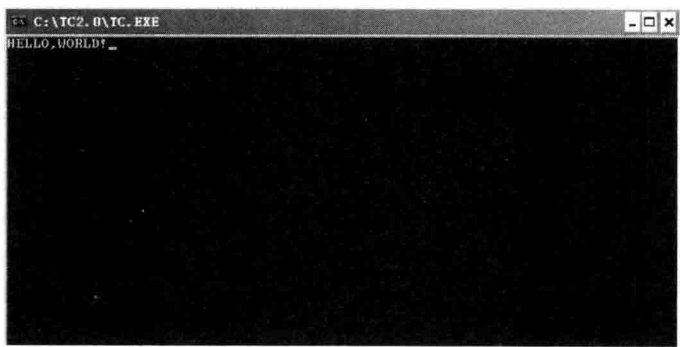


图 1-6 运行结果界面

1.2.1 C 语言的编写风格

一个编写规范的程序便于人们阅读、交流与调试。可读性好有助于人们对算法的理解，晦