



普通高等教育“十一五”国家级规划教材



C/C++ YU SHUJUJIEGOU (DI 4 BAN)
C/C++ 与数据结构 (第4版)

(上册)
(SHANGCE)

王立柱 王春枝 主编
Wang Lizhu Wang Chunzhi



清华大学出版社

C/C++与数据结构（第4版）

（上册）

王立柱 王春枝 主 编

叶志伟 欧阳勇 副主编

清华大学出版社
北京

内 容 简 介

本书主要包括 C、C++ 和 C++ 类模板三部分。C 语言部分介绍指针、数组、函数、字符串、结构体、顺序表、链表、文件和二维数组。C++ 部分介绍顺序表类、String 类、非线性结构与递归、继承和动态绑定、流与文件。C++ 类模板部分介绍向量类模板、链表类模板和适配器。本书既适用于 C 和 C++ 语言综合教学,又适用于程序设计类课程。

全书内容翔实、连贯,结构清晰、严谨,目标具体、明确,为高等院校计算机专业程序设计课程的整合指出了—个方向。本书不仅可以作为计算机及相关专业的程序设计与数据结构的教材,也可作为程序设计与数据结构爱好者的参考书。

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

C/C++ 与数据结构(上册)/王立柱,王春枝主编. —4 版.—北京:清华大学出版社,2016

ISBN 978-7-302-42204-4

I. ①C… II. ①王… ②王… III. ①C 语言—程序设计—高等学校—教材 ②数据结构—高等学校—教材
IV. ①TP312 ②TP311.12

中国版本图书馆 CIP 数据核字(2015)第 279017 号

责任编辑:白立军

封面设计:傅瑞学

责任校对:李建庄

责任印制:何 芊

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 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×230mm

印 张:27.5

字 数:548 千字

版 次:2002 年 3 月第 1 版

2016 年 1 月第 4 版

印 次:2016 年 1 月第 1 次印刷

印 数:1~2000

定 价:49.00 元

产品编号:067510-01

前 言

纵观短暂的计算机发展史,算法和数据结构这两个主要方面一直保持不变。发展演化的只是它们之间的关系,就是所谓的程序设计。——Stanley B. Lippman

当一门学科的实证的知识材料积累到相当丰富的时候,就需要依据这些材料的内在联系把它们条理化、系统化,以便更有效地把握和运用这些知识。因此,这门学科就走进了逻辑思维(即理论思维)的领域。

每一个中学毕业生都学过几何学,即来自《几何原本》的几何学。它就是对人类四百多年的几何知识材料进行系统化的结果。它从几个公理出发,按照形式逻辑规则,推导出所有几何定理。这便是著名的数学公理化方法,也称为形式逻辑。这种方法是把学科的知识材料系统化的一种有效手段。

但是,某一学科的公理化方法只有在这个学科成熟的时候才会出现。如果一个学科在不断的发展中,例如,计算机程序语言,尽管有机器语言、汇编语言、C语言、C++语言、C#和Java,但每年还有几十种程序语言出现,对这种学科的知识材料进行系统化,就需要另一种逻辑——辩证逻辑,因为辩证逻辑是从事物的发展和变化中来观察事物。

辩证逻辑的出现主要依赖于自然科学和工业的强大而日益迅猛的发展,例如,细胞、能量转换和进化论的发现。这些发现使人们不仅能够说明自然界中各个领域内的过程之间的联系,而且总的说来也能够说明各个领域之间的联系。

辩证逻辑的核心是对立统一规律:一切事物都存在矛盾,矛盾的双方既对立又统一,从而推动着事物的发展。

程序语言的矛盾实质上是程序设计的矛盾,这个矛盾是数据结构和算法。数据结构是一组有组织的数据和对数据的基本操作,算法是用基本操作表示的对数据进行处理有限步骤。而程序语言是表示某类数据结构的语言,因此数据结构和算法的矛盾,也是程序语言和算法的矛盾。

为什么说数据结构和算法是一对矛盾呢?因为没有不依赖数据结构的算法,也没有不支持算法的数据结构,这是它们的相互统一;数据结构是相对稳定的,就像程序语言是相对稳定的一样,而算法是越来越复杂的,当算法复杂到一定程度的时候,就要求数据结

构继续发展,这是它们的相互对立。

因为C和C++属于核心的程序语言,所以本书把它们和数据结构作为研究对象,对它们的知识进行系统化。本书不从概念出发,而从算法的需要出发,一个问题解决了,新的问题又出现了,一个问题的链条,对应一个解答的链条;每一个问题都是程序设计中的问题,每一个问题的解决都产生新的程序,并产生新的概念;因此,从C到C++,再到C++标准模板库,其概念都是在针对具体问题而精心设计的程序链条中,沿着必然的发展过程,以一个扩展一个、一个转换一个的方式彼此联系着。在这个过程中,你感受最深的不是有一个高高在上的思想者的存在,他要你做什么,而是实实在在的你个人的存在,你知道要做什么,你可以清楚地描述出每一个问题是如何产生又如何解决的,而且这个过程不因人而异。

本书的脉络如下。

(1)第1章 机器语言模式。机器语言并不是本书要具体学习的内容,而仅仅是作为全书的逻辑起点,在这个起点上,计算机仅仅是可以逐条执行机器指令的工具。

为此,本章仅用很少的、虚拟的机器指令构成一台计算机,同时通过执行一个简单的求和程序,引入存储器、地址、寄存器、程序计数器等基本概念,它们属于计算机体系结构的内容。

例1.1和例1.2阐述了数据结构和算法在机器语言层面上的相互作用,在这个过程中产生的程序入口地址、子程序调用、断口地址、现场保护、栈等基本概念。

机器语言的局限性首先是不易阅读,克服这个局限性产生了汇编语言和汇编程序。

机器语言的另一个局限性是算法描述力不足,一个简单的算法就需要很多条机器指令,为了克服这个局限性,产生了高级语言。

(2)第2章 C语言模式。C语言主要用基本数据类型表示数据结构,基本操作的形式主要是运算符表达式。每一个运算符表达式都对应一个机器语言程序,这是数据结构和算法在不同层面上的相互转换。

(3)第3章 函数。函数是对运算符表达式的扩展,即对运算符表达式的局限性的克服。程序设计由此进入过程化。

(4)第4章 一维数组和指针。由处理一个对象扩展到处理一组对象,即数组,使对象的访问方式由基于对象名称的直接访问方式发展到基于对象指针的间接访问方式。

(5)第5章 C字符串。C字符串不仅是一个构造类型,也是C语言提供的过程化程序设计的范例。

(6)第6章 结构体、联合体和枚举。C语言提供了半类型自定义机制。所谓半类

型,是指因为不能封装和不能深复制等原因,所以不能和基本类型一样使用的自定义类型。

(7) 第7章 顺序表。顺序表是改进的数组。

(8) 第8章 链表。顺序表是线性连续结构,链表是线性非连续结构,与顺序表相比,链表在插入和删除操作方面提高了效率。

(9) 第9章 C的流与文件。每一个文件都对应一个指针,在标准文件的读写函数中,这个指针被隐藏了。把隐藏的指针显式地表示出来,通过类比,轻松地进入磁盘文件的读写操作。

(10) 第10章 二维数组和指针。它是一维数组和指针的自然推广。

(11) 第11章 从C到C++。顺序表暴露出C语言固有的局限性,由此产生了C++的基本概念,逐一克服了这些局限性。

(12) 第12章 顺序表类。综合运用第11章的C++基本概念,把C顺序表转换为C++顺序表类。

(13) 第13章 String类。把C字符串转换为C++的String类。

(14) 第14章 Date类。把C结构体Date扩展为C++的Date类。

(15) 第15章 非线性结构与递归。把递归与树形结构联系起来讲解,是本书的特色之一。

(16) 第16章 继承和多态性。由C过程化程序设计进入C++面向对象程序设计。

(17) 第17章 向量类模板。把顺序表类扩展为向量类模板。

(18) 第18章 链表类模板和适配器。把C链表扩展为C++链表类模板,并以此为底层结构,生成链队列和链栈。

(19) 第19章 C++综合设计实例。

(20) 第20章 C++的流与文件。把C的文件读写函数转换为C++的文件读写函数。

(21) 第21章 名字空间。

编者

2015年9月

目 录

第 1 章 机器语言模式	1
1.1 模拟机器指令集与程序设计举例	1
1.2 机器语言的局限性	7
问题与练习	8
第 2 章 C 语言模式	9
2.1 基于基本类型的编程模式	9
2.2 基本数据类型	19
2.2.1 整型	19
2.2.2 实型	21
2.2.3 字符型	22
2.3 运算符和表达式	25
2.3.1 自增、自减运算符和表达式	25
2.3.2 复合赋值运算符和表达式	26
2.3.3 条件表达式和逗号表达式	26
2.3.4 关系运算符和逻辑运算符	27
2.3.5 运算符优先级	29
2.4 类型转换	29
2.5 程序流程控制结构	30
2.5.1 if-else 语句	31
2.5.2 switch-case 语句	32
2.5.3 break 语句和 continue 语句	34
问题与练习	35

第3章 函数	38
3.1 函数自定义与调用	38
3.2 函数声明与定义	43
3.3 函数与变量的存储类别	44
3.3.1 自动局部变量	45
3.3.2 静态局部变量	48
3.3.3 外部变量	49
3.4 函数应用设计举例	51
3.4.1 阶乘累加	51
3.4.2 求 π 的近似值	52
3.4.3 求最大公约数	53
3.4.4 判断质数	54
3.4.5 数制转换	55
3.5 模块化程序设计	56
3.5.1 全局外部函数	57
3.5.2 静态外部函数	58
3.5.3 全局外部变量	59
3.5.4 静态外部变量	60
3.6 编译预处理	61
3.6.1 无参宏指令	61
3.6.2 带参宏指令	62
3.6.3 条件编译指令	64
3.6.4 文件包含指令	66
问题与练习	68
第4章 一维数组和指针	70
4.1 指针和指针传递	70
4.2 一维数组和指针	75
4.2.1 一维数组	75
4.2.2 指向一维数组的指针	78
4.2.3 数组类型和数组首元素类型	81

4.3	const 型指针	83
4.4	动态数组	86
4.5	数组和指针应用举例	90
4.5.1	Josephus 问题	90
4.5.2	选择排序	93
4.5.3	起泡排序	96
4.5.4	划分数组元素	98
4.5.5	删除数组中的重复数据	101
4.5.6	筛法求质数	102
4.5.7	顺序搜索和二分搜索	104
4.6	索引和指针	107
4.7	指针和左值	108
4.8	函数指针	108
	问题与练习	109
第 5 章	C 字符串	111
5.1	字符串常量和字符串变量	111
5.2	字符串基本操作函数原型	117
5.3	字符串基本操作函数实现	118
5.4	字符串基本操作函数的补充	122
5.4.1	取子串	123
5.4.2	子串插入	125
5.4.3	子串删除	127
5.4.4	字符查找	128
5.5	模式匹配	129
	问题与练习	131
第 6 章	结构体、联合体和枚举	133
6.1	结构体	133
6.1.1	结构体定义	133
6.1.2	结构体变量和 typedef 名字	134

6.1.3	结构体变量的初始化和赋初值	135
6.1.4	结构体数组	136
6.1.5	结构体的嵌套	138
6.1.6	结构体返回值和指针传递	139
6.1.7	数组和含有数组的结构体变量	140
6.2	联合体	142
6.3	枚举	145
6.4	结构体应用设计举例	147
6.4.1	模拟洗牌	147
6.4.2	Date 结构体	149
6.4.3	三天打鱼,两天晒网	153
	问题与练习	154
第7章	顺序表	158
7.1	数组的局限性	158
7.2	顺序表声明与实现	159
7.2.1	顺序表声明	160
7.2.2	顺序表实现	164
7.3	索引和指针	169
7.4	数据抽象和封装	171
	问题与练习	171
第8章	链表	173
8.1	链表的结构分析	173
8.2	链表的声明和实现	179
	问题与练习	185
第9章	C 的流与文件	186
9.1	文件指针	186
9.2	文件打开与关闭	187
9.3	文件的读写	191

9.3.1	字符的读写	191
9.3.2	字符串的读写	193
9.3.3	无格式读写	194
9.3.4	格式读写	197
9.3.5	文件的随机访问	199
	问题与练习	201
第 10 章	二维数组和指针	204
10.1	二维数组和指针	204
10.2	二维数组和一维数组	211
10.3	马鞍点	213
10.4	指针数组和二级指针	215
10.5	指针数组与二维数组	217
	问题与练习	219
第 11 章	从 C 到 C++	221
11.1	C 语言的固有局限性	221
11.2	内联函数	224
11.3	运算符重载和函数重载	225
11.3.1	运算符重载	225
11.3.2	函数重载	227
11.4	引用型	230
11.4.1	概念的由来	230
11.4.2	引用型及其应用	233
11.5	函数模板	235
11.6	提取符和插入符	237
11.7	默认参数	239
11.8	深入讨论——函数模板实例化中的问题	241
	问题与练习	242

第 12 章 顺序表类	243
12.1 从 C 顺序表到 C++ 顺序表类	243
12.2 new 和 delete 操作符	249
12.3 需要增加、删除和修改的成员函数	250
12.4 顺序表类的声明和实现	258
12.5 类模板	259
12.6 基本类型赋值形式的扩展	264
问题与练习	265
第 13 章 String 类	266
13.1 String 类的声明	266
13.2 String 类的实现	269
13.2.1 构造函数和析构函数	269
13.2.2 成员赋值运算符	271
13.2.3 成员转换	272
13.2.4 串连接	274
13.2.5 关系运算	278
13.2.6 求子串	279
13.2.7 子串插入	280
13.2.8 子串删除	284
13.2.9 下标运算符	285
13.2.10 字符查找	285
13.2.11 输入输出	287
13.3 模式匹配	289
13.4 String 类的深入讨论	291
13.4.1 转换赋值运算符函数的替代	291
13.4.2 成员函数“类串+C 串”的替代	291
13.4.3 explicit 修饰符	292
问题与练习	293

第 14 章 Date 类	295
14.1 Date 类的声明	295
14.2 Date 类的实现	299
14.3 静态数据成员和静态成员函数	304
14.4 封装的典型应用	307
问题与练习	309
第 15 章 非线性结构与递归	310
15.1 树形结构与递归	310
15.2 C++ 递归函数	315
15.3 汉诺塔问题	316
15.4 快速排序	320
15.5 八皇后	321
问题与练习	326
第 16 章 继承和多态性	327
16.1 构造函数的参数初始化表	327
16.2 继承	330
16.3 受保护成员	332
16.4 多态性和虚函数	333
16.5 虚析构函数	337
16.6 纯虚函数和抽象类	338
问题与练习	342
第 17 章 向量类模板	344
17.1 向量类模板的声明和实现	344
17.2 函数对象	351
问题与练习	354

第 18 章	链表类模板和适配器	356
18.1	链表类模板 List	356
18.2	链表和链表类模板的代码对比	368
18.3	适配器	371
18.3.1	链栈	371
18.3.2	链队列	372
18.3.3	优先级链队列	373
	问题与练习	374
第 19 章	C++ 综合设计实例	375
19.1	中缀表达式求值	375
19.2	事件驱动模拟	380
	问题与练习	391
第 20 章	C++ 的流与文件	392
20.1	格式化输入输出	393
20.1.1	设置流的格式化标志	393
20.1.2	格式输出函数	395
20.1.3	操作算子	396
20.2	文件的读写	399
20.2.1	字符读写函数	400
20.2.2	字符串读写函数	402
20.2.3	无格式读写函数	402
20.2.4	格式读写	404
20.2.5	随机访问	406
20.3	文件错误处理	407
	问题与练习	408
第 21 章	命名空间	409
21.1	命名空间的定义	409

21.2	using namespace 语句	410
21.3	命名空间的成员	412
21.4	命名空间的别名	414
	问题与练习	414
附录 A	命名规则	415
附录 B	常用的 ANSI C 标准库函数	416
参考文献	423

第 1 章 机器语言模式

机器指令集是对实际处理硬件的抽象。使用这个抽象,机器代码程序表现得就像它是运行在一个一次只执行一条指令的处理器上。——Randal E. Bryant

机器语言(machine language)是一个机器指令集,又称为**机器码**(machine code),它们直接联系着计算机硬件,每一条机器指令都是计算机可以直接执行的。学习机器语言程序设计,需要深入了解计算机体系结构。但是本章的重点不是机器语,因此基本上不涉及计算机体系结构,只用若干条虚拟的机器指令集来模拟计算机,并通过几个程序来了解数据结构和算法这对矛盾最初是如何表现的。

1.1 模拟机器指令集与程序设计举例

在机器语言程序设计中,程序员亲自安排数据的存储,然后调用机器指令处理数据。

一条机器指令包含操作码和操作数两部分,前者是操作内容,后者是数据所在的存储单元地址,有时直接就是数据。例如,01H 3000H 是一条机器指令,其中 01H 是操作码,3000H 是操作数地址,具体意思是:取出地址为 3000H 存储单元中的数据,存入到 CPU 的寄存器 A 中。其中,H 表示操作码 01 和地址 3000 都是用十六进制表示的二进制数。1 位十六进制数和 4 位二进制数的对应关系如表 1.1 所示。

表 1.1 1 位十六进制数和 4 位二进制数的对应表

十六进制	二进制	十六进制	二进制	十六进制	二进制	十六进制	二进制
0	0000	4	0100	8	1000	C	1100
1	0001	5	0101	9	1001	D	1101
2	0010	6	0110	A	1010	E	1110
3	0011	7	0111	B	1011	F	1111

计算机存储器中存储的是一系列连续的字节,每一个字节有8位。指令01H 3000H需要3个字节。

表1.2是一台模型计算机的机器指令集。

表 1.2 一台模型计算机的机器指令集

指令名称	机器指令		指令说明
	操作码	操作数	
取数	01H	N	$A \leftarrow (N)$: 从地址为 N 的单元读取数据,存入寄存器 A
直接取	10H	C	$A \leftarrow C$: 把数据 C 存入寄存器 A
存数	02H	N	$(N) \leftarrow A$: 从寄存器 A 中读取数据,存入地址为 N 的单元
加法	03H	N	$A \leftarrow A + (N)$: 将地址为 N 的单元中的数据经过某一个寄存器和寄存器 A 中的数据相加,结果存入 A
直接加	30H	C	$A \leftarrow A + C$: 将数据 C 经过某一个寄存器和寄存器 A 中的数据相加,结果存入 A
乘法	04H	N	$A \leftarrow A \times (N)$: 将地址为 N 的单元中的数据经过某一个寄存器和寄存器 A 中的数据相乘,结果存入 A
转移	06H	N	$PC \leftarrow N$: 用地址 N 更新程序计数器 PC 的值
停机	00H		停机

例 1.1 编程计算 $z = x + y$ 。

算法步骤:

- (1) 从 x 中读数。
- (2) 从 y 中读数,计算 $x + y$ 。
- (3) 把求和结果存到 z 。

首先把 x 、 y 和 z 安排在存储器数据区,地址依次为 3000H、3001H 和 3002H,如图 1.1(a)所示。

算法的步骤可以用表 1.2 中的机器指令表示,其对应关系如表 1.3 所示。

机器指令位于存储器的程序区,除停机指令之外,其他指令都占3个字节,机器指令首字节地址称为指令地址。与算法步骤对应的4条机器指令的地址依次是 2000H、2003H、2006H 和 2009H,如图 1.1 所示。CPU 中的程序计数器 PC 存储机器指令地址,CPU 不断从 PC 中提取指令执行,直到停机为止。每执行一条指令,PC 都会指向下一条