

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

C语言程序设计

彭慧卿 邢振祥 主编

清华大学出版社

013028874

21 世纪普通高校计算机公共课程

TP312C-43
796

C语言程序设计

彭慧卿 邢振祥 主编



清华大学出版社

TP312C-43
796



北航

C1638079

内 容 简 介

本书是将C语言作为入门语言的程序设计类课程编写的教材,以培养学生程序设计的基本能力为目标。全书共分为10章:C语言概述,C语言基本数据类型、运算符及表达式,简单程序设计,选择结构程序设计,循环结构程序设计,数组,函数与预处理,指针,结构体、共用体和枚举,文件。

本书集作者多年的“C程序设计”课程的教学经验,全书组织精练、内容全面、条理清晰;每章附有较多的图表和程序流程图,讲述力求理论联系实际、循序渐进、深入浅出、通俗易懂,注重培养读者分析问题和程序设计的能力,以及良好的程序设计风格和习惯,具有较强的实用性。

本书可作为高等学校程序设计语言课程的教学用书,也可作为计算机等级考试的参考书,还可作为对C语言程序设计感兴趣的读者的自学用书。

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语言程序设计/彭慧卿等主编. —北京:清华大学出版社,2013.2

(21世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-31268-0

I. ①C… II. ①彭… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第008198号

责任编辑:付宏宇 薛 阳

封面设计:何凤霞

责任校对:梁 毅

责任印制:王静怡

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:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:17.25

字 数:429千字

版 次:2013年2月第1版

印 次:2013年4月第2次印刷

印 数:3001~5000

定 价:29.80元

产品编号:049574-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选编。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人:梁颖 liangying@tup.tsinghua.edu.cn

前言

C 语言是一种结构化程序设计语言,它功能丰富、表达能力强、使用灵活、应用面广、可移植性好,既具有高级语言的特点,又具有低级语言的许多功能,既可以用来编写系统软件,也可以用来编写应用软件。C 语言作为一门较为通用的编程语言,得到越来越广泛的应用。目前,C 语言程序设计被许多高校列为程序设计课程的首选,全国及各省市的计算机等级考试等都将 C 语言列入了考试范围。

本书是针对非计算机专业初学者的特点编写的,把提高编程能力、阅读程序的能力放在重要地位,避免在程序设计教学过程中陷入学习程序设计语言繁杂的语法和格式,而是将主要精力集中在所要解决的实际上,从知识点以及具体问题出发,重点讲述如何通过程序设计来解决问题。

本书是根据作者多年的教学经验和教学改革成果编写而成。全书以程序设计为主线,以编程应用为出发点,通过案例和问题引入内容,重点讲解程序设计的思想和方法,并介绍相关的语言知识,注重培养读者分析问题和程序设计的能力,注重培养良好的程序设计风格和习惯。本书讲述力求理论联系实际、循序渐进,全书通过大量例题验证语法和说明程序设计方法,并且为了使读者更好地掌握各章节内容,章末配有大量精选的习题。通过本书的学习和解题练习,读者既能迅速掌握 C 语言的基础知识,又能很快地学会 C 语言的编程技术,提高解决实际问题的能力。

为了更好地理解 C 语言编程的思想,将所学知识整体连贯,提高学生综合分析问题和解决问题的能力,为以后各种编程语言的学习打下坚实的基础,本教材安排了一个综合案例——学生成绩管理系统。综合案例的设计几乎涵盖了 C 语言的所有知识点,是对 C 语言的综合应用,是对所学知识的总结,旨在帮助读者掌握完整的课程脉络。

全书共分为 10 章,第 1 章为 C 语言概述,主要内容包括程序设计的概念、C 语言的发展、特点和 C 程序的组成和运行步骤;第 2 章为 C 语言基本数据类型、运算符及表达式,主要内容包括数据类型、数据类型转换、运算符与表达式;第 3 章为简单程序设计,主要内容包括算法的概述、数据的输入与输出、顺序程序设计;第 4 章为选择结构程序设计,主要内容包括关系运算符和关系表达式、逻辑运算符与逻辑表达式、if 语句和 switch 语句的使用;第 5 章为循环结构程序设计,主要内容包括复合赋值运算、自增自减运算、while 型循环、do-while 型循环、for 型循环、循环的嵌套、continue 和 break 语句及三种控制结构的综合应用;第 6 章为数组,主要内容包括一维数组、二维数组、多维数组、字符数组;第 7 章为函数与预处理,主要内容包括函数概念、函数定义、函数间的参数传递、函数调用、函数变量的作用域、变量的存储类型、编译预处理(文件包含、宏定义);第 8 章为指针,主要内容包括指针的含义、指针变量、指针与数组、指针与函数、指针与字符串、指向指针的指针;第 9 章为结构体、

共用体和枚举,主要包括结构体的定义和使用、共用体、枚举类型和使用 typedef 语句自定义数据类型;第 10 章为文件,主要包括 C 文件的概述、文件的打开与关闭、文件的读写、文件的定位。

本书中的程序都是在 Visual C++ 6.0 编译环境下进行调试的,在其他 C 语言环境下基本上都可以运行通过。

作者还编写了配套的《C 语言程序设计学习指导》一书,内容包括习题解答和实验指导。习题解答包括与教材配套的习题解答和解析、大量的练习与答案,以帮助读者巩固各章节知识点;实验指导共有 9 个实验,每个实验内容又分为基础型实验、设计型实验和提高型实验,供不同程度的读者选作,在每个实验中还包括实验指导、设计分析、操作指导和常见问题分析,以帮助读者更好地理解实验内容,高质量地完成实验;最后给出了实验思考题。

本书由天津城市建设学院彭慧卿副教授、邢振祥教授担任主编,天津城市建设学院彭慧卿编写第 7、9 章,刘琦编写第 1、5 章,李耀芳编写第 2、3 章,高晗编写第 4、10 章,戴华林编写第 6 章,戴春霞编写第 8 章。最后由彭慧卿统稿,邢振祥教授、孙莹光副教授审阅了全书并提出了宝贵意见。

在本书编写过程中,编者参考了大量有关 C 语言程序设计的书籍和资料,在此对这些参考文献的作者表示感谢。

由于编者水平有限,不足之处在所难免,恳请广大读者批评指正。

编 者

2012 年 11 月

目 录

第 1 章 C 语言概述	1
1.1 程序与程序设计语言	1
1.1.1 程序的基本概念	1
1.1.2 程序设计语言	2
1.2 C 语言的历史背景	4
1.3 C 语言的特性	5
1.4 C 语言程序的基本结构	5
1.4.1 C 语言程序的结构特点	5
1.4.2 程序设计风格	8
1.5 C 语言编译环境介绍	9
1.5.1 Visual C++ 6.0 的启动	9
1.5.2 编辑源程序	10
1.5.3 编译、连接和执行	11
1.5.4 关闭工作区	12
习题	12
第 2 章 C 语言基本数据类型、运算符及表达式	14
2.1 标识符	14
2.1.1 字符集	14
2.1.2 C 语言词汇	15
2.2 C 语言的数据类型	16
2.3 常量和变量	17
2.4 整型数据	18
2.4.1 整型常量	18
2.4.2 整型变量	19
2.5 实型数据	20
2.5.1 实型常量	20
2.5.2 实型变量	21
2.6 字符型数据	22
2.6.1 字符常量	22

2.6.2 字符变量	24
2.7 运算符和表达式	25
2.7.1 简单赋值运算符及其表达式	25
2.7.2 算术运算符及其表达式	27
2.7.3 数据类型转换	28
2.7.4 位运算符及其表达式	31
习题	35
第3章 简单程序设计	38
3.1 算法	38
3.1.1 算法的概念	38
3.1.2 算法的描述	38
3.2 结构化程序设计	41
3.2.1 C 语言程序结构分类	42
3.2.2 C 语言语句分类	42
3.3 数据输入输出	43
3.3.1 格式输出函数 printf	44
3.3.2 格式输入函数 scanf	50
3.4 字符数据的输入输出	53
3.5 顺序结构程序设计	54
习题	56
第4章 选择结构程序设计	60
4.1 关系运算符与关系表达式	60
4.1.1 关系运算符	60
4.1.2 关系表达式	61
4.2 逻辑运算符与逻辑表达式	61
4.2.1 逻辑运算符	62
4.2.2 逻辑表达式	62
4.3 if 语句	64
4.3.1 单分支选择结构 if 语句	64
4.3.2 双分支选择结构 if-else 语句	66
4.4 if 语句的嵌套	67
4.5 switch 语句	71
4.6 选择结构应用举例	74
习题	76
第5章 循环结构程序设计	80
5.1 循环的概念	80

5.2	for 语句	81
5.2.1	for 语句的结构	81
5.2.2	复合赋值运算	84
5.2.3	自增和自减运算	85
5.2.4	逗号运算	86
5.3	while 语句	88
5.4	do-while 语句	89
5.5	如何跳出循环结构	91
5.6	循环的嵌套	93
5.7	三种循环的比较	95
5.7.1	循环语句的选择	95
5.7.2	无限循环	98
5.8	循环结构应用实例	98
	习题	101
第 6 章	数组	106
6.1	一维数组	106
6.1.1	一维数组的定义	107
6.1.2	一维数组的初始化	108
6.1.3	一维数组元素的引用	109
6.1.4	一维数组程序举例	110
6.2	二维数组	115
6.2.1	二维数组的定义	115
6.2.2	二维数组元素的引用	116
6.2.3	二维数组的初始化	117
6.2.4	二维数组程序举例	119
6.3	字符数组	122
6.3.1	字符数组的定义与初始化	122
6.3.2	字符串	123
6.3.3	字符串的操作	123
6.3.4	字符串处理函数	125
6.3.5	字符数组程序举例	128
	习题	132
第 7 章	函数与预处理	137
7.1	模块化程序设计方法	137
7.2	库函数介绍	139
7.3	用户自定义函数的基本概念	140
7.3.1	函数的定义	140

7.3.2	形参和实参	142
7.3.3	函数的返回值	143
7.4	函数的调用	145
7.4.1	函数的调用形式	145
7.4.2	函数的调用过程	146
7.4.3	函数的声明	147
7.5	函数的参数传递	149
7.5.1	值传递	149
7.5.2	地址传递	151
7.6	函数的嵌套调用和递归调用	153
7.6.1	函数的嵌套调用	153
7.6.2	函数的递归调用	154
7.7	变量的作用域和存储类别	158
7.7.1	变量的作用域	158
7.7.2	变量的存储类别	161
7.8	预处理命令	164
7.8.1	宏定义	164
7.8.2	文件包含	169
7.9	大程序的组成	170
7.9.1	C 程序的组成	170
7.9.2	源文件间的通信	171
7.10	应用举例	173
	习题	176
第 8 章	指针	182
8.1	变量的地址和指针	182
8.2	指针变量	183
8.2.1	指针变量的定义和初始化	183
8.2.2	指针的运算	184
8.3	指针和数组	187
8.3.1	指针和一维数组	187
8.3.2	指针和二维数组	190
8.4	指针与字符串	191
8.4.1	字符指针	191
8.4.2	字符指针与字符数组	192
8.5	指针与函数	194
8.5.1	指针作为函数的参数	194
8.5.2	数组名与指针作为函数参数的比较	196
8.5.3	指针型函数	199

8.5.4	指向函数的指针·····	200
8.6	指向指针的指针和指针数组·····	202
8.6.1	指向指针的指针·····	202
8.6.2	指针数组·····	203
8.6.3	行指针·····	205
8.7	main 函数的形参·····	207
	习题·····	209
第 9 章	结构体、共用体和枚举·····	215
9.1	结构体类型的定义·····	215
9.2	结构体变量的定义和使用·····	216
9.2.1	结构体变量的定义·····	216
9.2.2	结构体变量的引用·····	217
9.2.3	结构体变量的初始化·····	219
9.2.4	结构体变量作为函数参数·····	221
9.3	结构体数组·····	222
9.3.1	结构体数组的定义及初始化·····	222
9.3.2	结构体数组应用举例·····	223
9.4	结构体指针·····	225
9.4.1	指向结构体变量的指针·····	225
9.4.2	指向结构体数组的指针·····	227
9.4.3	结构体指针作为函数参数·····	228
9.5	共用体类型·····	228
9.5.1	共用体的概念·····	228
9.5.2	共用体类型定义和变量定义·····	229
9.6	枚举类型·····	231
9.6.1	枚举类型变量的定义·····	231
9.6.2	枚举类型的应用·····	232
9.7	用 typedef 自定义数据类型·····	233
	习题·····	234
第 10 章	文件·····	238
10.1	文件概述·····	238
10.1.1	文件的概念·····	238
10.1.2	文件的分类·····	239
10.1.3	缓冲文件系统·····	239
10.1.4	文件指针·····	240
10.1.5	文件的操作顺序·····	240
10.2	文件操作·····	241

10.2.1 文件的打开和关闭	241
10.2.2 文件的顺序读/写	243
10.2.3 文件的定位与随机读写	249
10.3 文件应用综合实例	251
习题	254
附录 A 标准字符与 ASCII 代码对照表	256
附录 B 运算符的优先级和结合性	257
附录 C C 常用库函数	259
参考文献	262

在人们使用计算机的过程中,要使计算机按人们预先安排的步骤进行工作,就要解决人与计算机的交流问题。人与计算机进行交流的语言,称为程序设计语言。C 语言是国内外广泛流行的高级程序设计语言,既可用它编写应用软件,也能用它编写包括操作系统在内的系统软件。C 语言既具有高级语言的特点,又具有汇编语言的功能。用 C 语言编写的程序具有良好的可移植性和较高的执行效率。随着计算机的广泛使用,C 语言在各个领域的应用也越来越广泛。

本章从程序设计的角度,结合 C 语言的发展和特点,介绍有关程序设计的基本概念,以及 C 语言程序的基本结构等内容。

1.1 程序与程序设计语言

1.1.1 程序的基本概念

要使计算机为人类完成各种各样的工作,就必须让它执行人们预先设计好的相应程序。

1. 程序

所谓程序,实际上是用计算机语言描述的某一问题的解题步骤,是符合一定语法规则的符号序列。它表达了人们解决问题的过程,通过在计算机上运行程序,向计算机发出一系列指令,告诉计算机要处理什么以及如何处理,计算机便可按人们的要求解决特定问题。

一个程序一般应包含以下两方面内容:一是对数据的描述,在程序中有指定数据的类型和数据的组织形式,即数据结构;二是对操作步骤的描述,也就是算法。

2. 程序设计

程序设计的目的是用计算机解决问题。所谓程序设计就是把解题步骤用程序设计语言描述出来的工作过程。用计算机解决问题大体上经过以下几个步骤。

1) 问题分析

用计算机来解决问题,首先应通过对问题的分析,确定在解决这个问题过程中要做些什么。分析问题、弄清楚要解决的问题并给出问题的明确定义是解决问题的关键。

2) 算法设计

在弄清要解决的问题之后,就要考虑如何解决它,即如何做。这就是算法设计。算法设计分为以下两个步骤。

(1) 确定数据结构。

根据任务提出的要求、指定的输入数据和输出结果,对问题进行抽象,抽取出能够反映本质特征的数据并对其进行描述,确定存放数据的数据结构。

(2) 确定算法。

针对设计好的存放数据的数据结构考虑如何进行操作以获得问题的结果,即确定解决问题、完成任务的步骤。

3) 编写程序

根据确定的数据结构和算法,使用选定的程序设计语言编写程序代码,简称编程。

4) 编译运行和调试

通过对程序的调试消除由于疏忽而引起的语法错误或逻辑错误;用各种可能的输入数据对程序进行测试,使之对各种合理的数据都能得到正确的结果,对不合理的数据能进行适当的处理。

程序设计既是一门科学,又是一门艺术。就像练习写作一样,必须不断地编程实践并且大量阅读他人的程序,积累经验,才能形成良好的程序设计风格,不断提高程序设计能力。

1.1.2 程序设计语言

程序设计语言就是用户用来编写程序的语言,它是人与计算机之间交换信息的工具。当今程序设计语言发展非常迅速,新的程序设计语言层出不穷,其功能越来越强大。根据程序设计语言与计算机硬件的联系程度可分为机器语言、汇编语言和高级语言三类。

1. 机器语言

机器语言是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合。机器语言具有灵活、直接执行和速度快等特点。不同型号的计算机其机器语言互不兼容,按照一种计算机的机器指令编制的程序,不能在另一种计算机上执行。

要使用机器语言编写程序,编程人员必须熟记所用计算机的全部指令代码和代码的含义,程序员需要自己处理每条指令和每一数据的存储分配和输入输出,还必须记住编程过程中每一步所使用的存储单元处在何种状态。这是一件十分烦琐的工作,编写程序花费的时间往往是实际运行时间的几十倍或几百倍。而且,编出的程序全是些 0 和 1 的指令代码,可读性差,且容易出错。现在,除了计算机生产厂家的专业人员外,绝大多数的程序员已经不使用机器语言了。

2. 汇编语言

为便于理解与记忆,人们采用一些“助记符号”来表示机器语言中的机器指令,这样便形成了汇编语言。助记符一般都是采用一个操作的英文字母的缩写,如用 ADD 表示加法。

不同类型的计算机对应的指令系统也是不同的。由于汇编语言采用了助记符,因此,它比机器语言直观,容易理解和记忆。用汇编语言编写的程序也比机器语言程序易读、易检查、易修改。但是,计算机不能直接识别源程序,必须由一种专门的翻译程序将汇编语言源程序翻译成机器语言程序后,计算机才能识别并执行。另外,汇编语言除了可读性比机器语言好以外,同样也存在机器语言的缺点,尤其是描述问题的方式与人们习惯相距太远。

3. 高级语言

机器语言和汇编语言都是面向机器的语言,一般称为低级语言。低级语言对机器的依

赖性太大,人们在使用它们设计程序时,要求对计算机比较熟悉。用它们开发的程序通用性差,普通的编程者很难胜任这一工作。

为了克服低级语言这一缺点,随着计算机技术的发展以及计算机应用领域的不断扩大,从20世纪50年代中期开始逐步发展了面向问题的程序设计语言,称为高级语言。高级语言与具体的计算机硬件无关,其表达方式更接近人类自然语言的表述习惯,具有很高的可读性。高级语言的一条语句通常对应于多条机器指令,所以用高级语言编写程序要比低级语言容易得多,并大大简化了程序的编制和调试,使编程效率得到大幅度的提高。高级语言的显著特点是独立于具体的计算机硬件,通用性和可移植性好。

高级语言是接近于自然语言和数学语言,在一定程度上与具体计算机无关的符号化语言。用高级语言描述的算法代码是一种符号化的语句序列,也称为源程序。

高级语言易学易用,程序易理解、调试、修改和移植,但大部分语言不支持对硬件的直接操作,需要翻译成等价的指令序列后才能由计算机执行。翻译方式分为两种:编译方式与解释方式。

(1) 编译方式。用相应的编译程序(compiler,或称为编译器)对高级语言描述的源代码进行若干次扫描后生成目标代码。但目标代码并不能直接执行,还需要经过连接、装配成可执行指令序列代码后才能运行并获得结果。虽然编译方式实现复杂,但相对而言能产生效率较高的目标代码,编译一次,运行多次。这种翻译方式适合结构复杂、要求运行效率高的应用程序开发。

(2) 解释方式。用解释程序(interpreter)对源代码逐句扫描、处理、执行后直接获得结果。解释方式实现简单但效率低,同一代码每次运行都要进行解释。一般来说,交互式程序设计语言采用解释方式。

程序设计语言规定了一组记号和一组规则,用于描述或书写计算机程序。无论编译方式还是解释方式,用高级语言编写的程序,运行效率一般低于低级语言代码。

尽管计算机程序设计语言的差别很大,但无论哪种语言,其基本语言成分都可归结为4大类:用以刻画程序所处理的数据对象的值、存储、类型的数据成分,用以规定程序设计中所能进行的运算(如算术运算、逻辑运算、集合运算等)成分,用以控制程序执行流程的控制成分,用以表达程序中数据输入输出的传输成分。

常用的计算机高级语言包括以下几种。

(1) FORTRAN 语言(FORmula TRANslation)。它是最早出现的高级语言之一,广泛用于科学计算和数据处理。它第一次允许程序员用数学形式的语句来编写程序,例如 $A = B + C$ 。FORTRAN 最早出现在1957年,随着计算机技术的发展,先后出现了FORTRAN77、FORTRAN80、FORTRAN95等版本,其应用范围也不断扩大,至今仍是进行数学计算的主要语言。

(2) BASIC 语言。起源于20世纪60年代初期的BASIC语言,吸收了近代语言的研究成果,已经发展为一种面向对象的程序设计和软件开发语言,广泛用于数值计算、自动控制系统中。

(3) Pascal 语言。20世纪70年代初期,N. Wirth设计了Pascal语言,它是一种结构化程序设计语言。由于其鲜明的结构化及严格的语法规约,比较适合教学。

(4) C 语言。最初C语言是为了编写UNIX操作系统而设计的。它是高级程序设计语

言,又具有汇编语言的一些优点,其应用范围很广。C 语言是一种结构化、模块化、可编译的通用程序设计语言,广泛应用于系统软件和应用软件的开发。

(5) C++。C++是由 C 语言发展而来的,与 C 兼容。用 C 语言编写的程序基本上可以不加修改地用于 C++。从 C++ 的名字可以看出它是 C 的超集。C++既可用于面向过程的结构化程序设计,也可用于面向对象的程序设计,是一种功能强大的混合型程序设计语言。C++在原来面向过程的机制基础上,对 C 语言的功能作了不少扩充,并增加了面向对象的机制。在 C++产生之后,又出现了 Borland C++、C++ Builder 和 Visual C++ 等针对 C++ 语言的集成开发环境。而以 Visual C++ 为代表的集成开发环境在 C++ 基础上增加了许多其他特性。本书中的所有例子均是在微软的 Visual C++ 6.0 环境下编写、调试和运行测试的。

(6) COBOL(Common Business Oriented Language,面向商业的通用语言)。COBOL 是在 20 世纪 50 年代推出,是最早的数据处理语言,至今还用于一些数据处理的场合,特别是管理信息系统中。COBOL 允许用类似英语的简单句子来编写程序,所以它在商业、行政管理和数据处理方面得到了广泛应用。

1.2 C 语言的历史背景

C 语言是由 B 语言发展而来的,它的根源可以追溯到 ALGOL 60 程序设计语言,但因为 B 语言只有单一的字符类型、过于简单等原因而未能流行。1972—1973 年间,贝尔实验室的 Dennis. M. Ritchie 和 Brian. W. kernighan 对 B 语言进一步改进,设计出 C 语言。C 语言充分吸收了 BCPL 和 B 语言的优点(精练,接近硬件),同时也克服了它们的缺点(过于简单,数据无类型,功能较差等)。1973 年,两人合作用 C 语言改写了 UNIX 操作系统 90% 以上的内容,即 UNIX 第 5 版(以前的 UNIX 系统是两人用汇编语言编写的)。

随后,贝尔实验室对 C 语言进行了多次改进,1975 年,随着 UNIX 第 6 版公布,C 语言受到人们普遍关注,它的突出优点得到计算机界的认可。C 语言可以独立于 UNIX 操作系统和 PDP 机存在,并且可以移植到各种大、中、小、微型计算机上,这使得 UNIX 操作系统得到广泛推广。随着 UNIX 操作系统的日益广泛使用,C 语言也迅速发展。因此可以说,C 语言和 UNIX 是一对孪生兄弟,相辅相成。

1978 年,Dennis. M. Ritchie 和 Brian. W. kernighan 合作编写了经典著作《The C Programming Language》,书中详细阐述了 C 语言,人们把它称为标准 C,它是目前所有 C 语言版本的基础。1983 年,美国国家标准化协会(ANSI)对 C 语言问世以来的各种版本进行扩充,对 C 语言进行了标准化,即 ANSI C。现在流行的 C 语言编译环境比较多,如 Microsoft C、Turbo C、Quick C、Borland C、Visual C++ 等。

在 C 语言的基础上,1983 年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言,成为一种面向对象的程序设计语言。C++ 提出了一些更为深入的概念,它所支持的面向对象的概念容易将问题空间直接地映射到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性,掌握起来有一定难度。但是,C 是 C++ 的基础,C++ 语言和 C 语言在很多方面是兼容的。因此,掌握了 C 语言,再进一步学习 C++ 就能以一种熟悉的语法来学习面向对象的语言,从而达到事半功倍的目的。