

21世纪计算机专业大专系列教材

李大友 主编

C++语言 程序设计

徐孝凯 编著



清华大学出版社

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

21 世纪计算机专业大专系列教材

李大友 主编

C++ 语言程序设计

徐孝凯 编著

清华大学出版社

内 容 简 介

本书是李大友教授主编的 21 世纪计算机专业大专系列教材之一,主要介绍算法和流程图的基本概念,C++ 语言中常量、变量、表达式、语句、函数等的表示和作用,数组、字符串、指针、结构、联合、类、文件等数据类型的定义与使用,累加、统计、查找、排序、更新等运算的基本程序设计方法,C++ 程序的基本结构,Visual C++ 6.0 集成开发环境的使用等内容。本书具有丰富的程序例题、编程例题、练习题和上机题,能够使學生得到良好的程序设计训练。

本书是以系统学习计算机高级程序设计语言的读者为对象编写的,具有概念定义准确、叙述深入浅出、结构层次分明、分析问题透彻、章节安排有序、便于阅读和自学等特点。通过学习本书能够很好地掌握 C++ 语言的基本语法规则,具备分析和设计一般应用程序的能力。

本书可作为计算机专业大专教材,也可作为非计算机专业本科教材,还可供工程技术人员参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

C++ 语言程序设计/徐孝凯编著. —北京:清华大学出版社,2002

(21 世纪计算机专业大专系列教材/李大友主编)

ISBN 7-302-06068-1

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

中国版本图书馆 CIP 数据核字(2002)第 087169 号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

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

责任编辑: 徐跃进

印 刷 者: 北京昌平环球印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 23.75 **字数:** 545 千字

版 次: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

书 号: ISBN 7-302-06068-1/TP·3620

印 数: 0001~6000

定 价: 29.00 元

《21 世纪计算机专业大专系列教材》

编辑委员会名单

主 编 李大友

编 委 (排名不分先后)

刘乐善 (华中理工大学)

刘惠珍 (北京工业大学)

陈 明 (石油大学)

邵学才 (北京工业大学)

蒋本珊 (北京理工大学)

匙彦斌 (天津大学)

葛本修 (北京航空航天大学)

彭 波 (中国农业大学)

徐孝凯 (中央广播电视大学)

策划编辑 范素珍

序

这套教材是 21 世纪高等学校计算机专业大专系列教材。

我们从 1995 年开始组织《计算机专业大专系列教材》的编写。当时根据中国计算机学会教育委员会与全国高等学校计算机教育研究会联合推荐的《计算机学科教学计划 1993》的要求,组织了《计算机组成原理》等 13 本教材,并由清华大学出版社出版。这套教材出版后,受到了高等学校师生的广泛欢迎和好评。

上述教材的组织,主要是按《计算机学科教学计划 1993》的要求进行的。而 1993 教学计划主要是参照美国 IEEE 和 ACM《计算机学科教学计划 1991》并结合我国高等教育当时的实际情况制定的,反映的是 20 世纪 80 年代末计算机学科的发展状况。

计算机学科是一个飞速发展的新兴学科,发展速度之快可谓一日千里。近 10 年来,计算机学科已发展成为一个独立学科,计算机本身向高度集成化、网络化和多媒体化迅速发展。但从另一个方面来看,高等学校的计算机教育一直滞后于计算机学科的发展,特别是教材建设,由于受时间和软硬条件的限制,更是落后于现实需要,而大专层次的教材建设问题尤其严重。为了改变这种状况,高等学校的教育工作者和专家教授们应当仁不让地投入必要的时间和精力来完成这一历史使命。

为组织好这套教材,我们认真地研究了全国高等学校计算机专业教学指导委员会和中国计算机学会教育委员会联合推荐的《计算机学科教学计划 2000》和美国 IEEE 和 ACM 两个学会最新公布的《计算机学科教学计划 2001》。这两个教学计划都是在总结了从《计算机学科教学计划 1991》到现在计算机学科十年来发展的主要成果的基础上诞生的。它们所提供的指导思想和学科所涵盖的内容,不仅适合于大学本科,也适合大学专科的需求,关键在于要对其内容的取舍进行认真的研究。

在我国的《计算机学科教学计划 1993》和美国 IEEE 和 ACM 两个学会提出的《计算机学科教学计划 1991》中,根据当时的情况,只提出了 9 个主科目。而在《计算机学科教学计划 2001》中,根据学科的最新发展状况,提出了 14 个主科目,其中 13 个主科目为核心主科目。这 14 个主科目是:算法与分析(AL)、体系结构(AR)、离散结构(DS)、计算科学(CN)、图形学与可视化计算(GV)、网络计算(NC)、人机交互(HC)、信息管理(IM)、智能系统(IS)、操作系统(OS)、程序设计基础(PF)、程序设计语言(PL)、软件工程(SE)、社会、道德、法律和专业问题(SP),其中除 CN 为非核心主科目外,其他 13 个主科目均为核心主科目。

将美国 IEEE 和 ACM 的教学计划 2001 与 1991 计划进行比较可看出:在 1991 计划中,离散结构只是作为数学基础提出,未被列为主科目;而在 2001 计划中,不但列为主科

目,而且为核心主科目。可见,已将离散结构提升为本学科的基础。

在 1991 计划中,未提及网络计算,而在 2001 计划中,不但提出,而且被列为核心主科目,以适应网络技术飞速发展的需求。

图形学与可视化计算也是为适应发展需求新增的内容,并且列为主科目。

除此之外,2001 计划在下述 5 个方面做了增加或调整:

- 将程序设计语言引论调整为程序设计基础和程序设计语言两个核心主科目,显然,加强了对程序设计的要求。
- 将人-机通信调整为人机交互,反映了人-机通信的实质是人机交互。在图形界面迅速发展的今天,人机交互理论和方法的研究和应用变得十分重要。
- 将人工智能与机器人学调整为智能系统,拓宽了对智能系统的要求。
- 将数据库与信息检索调整为信息管理,因为后者不仅概括了前者,而且反映了数据库与信息检索的实质是信息管理。
- 将数值与符号计算调整为计算科学,更具有概括性。

总之,上述变化不仅更好地反映了计算机学科的发展现状,而且使 2001 教学计划具有更强的科学性和实用性。

由于这套系列教材主要面向的对象是计算机专业三年制大专(高职)学生,其培养目标也应属于高级技术人才的层次。他们既要有一定的理论基础(较本科弱),又要更强调实用性,要有明确的应用方向。我们将应用方向定位在信息管理和计算机网络两个方向。这两个应用方向占计算机应用总计的 90%以上。

在系列教材的内容取舍上,2001 教学计划的 14 门主科目中,我们概括了除智能系统、计算科学和社会、道德、法律和专业问题之外的其他 11 个主科目。在每个主科目中,我们都以其中的基本概念、基本理论和基本方法作为主线组织教材,使学生既能掌握基本的基础理论和方法,又能为他们进一步深造打下必要的基础;在信息管理和计算机网络技术两个应用方向上,他们的应用能力将得到加强。

根据上述指导思想,初步确定组织 20 本左右的教材供各高校选用。这些教材包括:《离散数学》、《计算机应用基础》、《计算机组织与结构》、《微机系统与接口技术》、《计算机网络与通信》、《网络管理技术基础》、《计算机网络系统集成技术》、《数据结构》、《操作系统原理》、《实用软件工程基础》、《数据库原理与应用》、《管理信息系统原理与应用》、《办公自动化实用技术》、《多媒体技术及其应用》、《Internet 技术及其应用》、《计算机维护技术》、《C 语言程序设计》、《Java 语言程序设计》、《C++ 语言程序设计》、《Visual Basic 程序设计》、《计算机英语》等。

系列教材并不是教学计划,由于各高校情况不同,培养方向的侧重面也不一样,因此教学计划也不会雷同。教材按系列组织,力图能够反映计算机学科大专层次的总体要求,同时采用大拼盘结构,各校可根据自身情况选择使用。例如,语言类教材,我们就准备了多本,各校可选择其中的一本或两本,其他依此类推。

这套教材均由高等学校具有丰富教学实践经验的老师编写。所编教材体系结构严谨、层次清晰、概念准确、理论联系实际、深入浅出、通俗易懂。相信一定能够得到专科院校计算机专业师生的欢迎。

全国高等学校计算机教育研究会副理事长
课程与教材建设委员会主任

李大友

2001.6

前 言

本书为 21 世纪计算机专业大专层次系列教材之一,是根据美国 IEEE 和 ACM 两个学会最新公布的《计算机学科教学计划 2001》的要求编写的。

C++ 语言是当前最流行和最实用的一种计算机高级程序设计语言,它具有丰富的数据类型和各种运算功能,带有庞大的函数库和类库,既支持面向过程的程序设计,又支持面向对象的程序设计,特别是 Visual C++ 6.0 版本已成为当前进行软件开发的主要工具之一。

正因为 C++ 语言在软件开发中的广泛使用,现在普通高校开设的程序设计语言课程正在由 PASCAL 和 C 语言向 C++ 语言过渡,以满足培养计算机应用人才的实际需求。

本书共包含 11 章。第 1 章为程序设计引论,主要介绍数制、编码、算法、流程图等基本知识,使读者初步建立起利用计算机解决问题的方法和思路。第 2 章为 C++ 语言概述,主要介绍 C++ 语言的基本框架、C++ 程序的基本结构,使用 Visual C++ 6.0 集成开发环境输入、编辑、编译、连接和运行程序的操作过程等内容,使读者对 C++ 语言概况和上机操作过程有所认识。第 3 章为数据类型和表达式,详细讨论了 C++ 语言中各种简单数据类型、各种常量的表示、变量的定义、各种运算符的含义、常用数学函数的使用等内容。第 4 章为流程控制语句,详细讨论了条件语句、情况语句、各种循环语句的格式、功能和编程应用举例。第 5 章为数组和字符串,详细讨论了一维数组、二维数组和字符数组(即字符串)的定义及在程序设计中的应用。第 6 章为指针,详细讨论了指针的概念、指针的运算、指针的作用、动态存储分配等内容。第 7 章为函数,详细讨论了函数的定义与调用、变量的作用域等内容。第 8 章至第 10 章分别为结构与联合、类与对象、类的继承与多态性,主要讨论了各种用户定义的数据类型的功能及在程序设计中的应用。第 11 章为 C++ 流,主要介绍了数据的输入输出格式和文件的输入输出操作。各章内容之间连贯有序,衔接自然,成为一个有机的整体。

C++ 语言系统庞大,知识点之间像一张网,错综复杂,如何能够按照一般的认知规律和教学的需要,把有关知识点按章节划分为前后有序的一个线性结构,由浅入深、由易到难、循序渐进地组织内容,并使之前后呼应、条理清楚、方便自学,是编写教材成功与否的关键。本人在这方面做了很大的努力,使读者能够轻松地学习到 C++ 语言的基本内容,为学好后续各门课程及软件应用开发打下良好的基础。

本书中所有的 C++ 函数和程序都在 Microsoft Visual C++ 6.0 集成开发环境下运行通过,可以确保它们是正确无误的。

本课程总学时应安排在 80~100 之间,其中讲授与上机学时之比应为 3:2 左右。若课时紧张,目录中带星号的内容可以不讲。

本书可作为计算机专业大专教材,也可作为非计算机专业本科或专科教材。

为了有利于任课教师和广大自学读者使用此书,书中各章所有习题的参考解答已放

在清华大学出版社的网页上(<http://www.tup.tsinghua.edu.cn>),供免费访问。

本书由徐孝凯主持编写,解季萍和朱崑两位老师编写了部分章节的内容。

由于本人水平有限,加之时间仓促,错误和不足之处在所难免,敬请广大读者批评指正。本人联系电话 010-64910302,电子邮件地址 xuxk@crtvu.edu.cn。

徐孝凯

2002年12月于北京

目 录

第 1 章 程序设计引论	1
1.1 数制	1
1.1.1 不同数制的表示.....	1
1.1.2 不同数制的转换.....	3
1.2 编码	5
1.2.1 ASCII 码.....	5
1.2.2 汉字区位码.....	6
1.3 算法	7
1.3.1 算法的概念.....	7
1.3.2 算法设计举例.....	8
1.3.3 结构化程序设计的概念	10
1.4 用流程图表示算法的基本结构.....	11
1.4.1 传统流程图	11
1.4.2 盒图	12
1.4.3 问题分析图	12
1.5 用流程图描述算法.....	13
习题	19
第 2 章 C++ 语言概述	20
2.1 引言.....	20
2.2 C++ 字符集	21
2.3 C++ 单词	22
2.4 C++ 语句	24
2.5 C++ 函数	26
2.6 C++ 程序	29
2.7 VC++ 6.0 集成开发环境简介	32
习题	38
第 3 章 数据类型和表达式	42
3.1 数据类型.....	42
3.2 常量.....	44
3.2.1 整型常量	45
3.2.2 字符常量	45

3.2.3	逻辑常量	47
3.2.4	枚举常量	47
3.2.5	实型常量	48
3.2.6	地址常量	49
3.3	变量	49
3.4	运算符和表达式	53
3.5	函数	61
	习题	66
第4章	流程控制语句	74
4.1	if 语句	74
4.2	switch 语句	78
4.3	for 语句	82
4.4	while 语句	92
4.5	do 语句	98
4.6	跳转语句	102
	习题	106
第5章	数组和字符串	117
5.1	数组的概念	117
5.2	数组的定义	118
5.2.1	一维数组	118
5.2.2	二维数组	122
5.2.3	使用 typedef 语句定义数组类型	127
5.3	数组的应用	129
5.3.1	数值计算	129
5.3.2	统计	132
5.3.3	排序	134
5.3.4	查找	137
5.4	字符串	140
5.4.1	字符串概念	140
5.4.2	字符串函数	143
5.4.3	字符串应用举例	146
	习题	150
第6章	指针	157
6.1	指针的概念	157
6.2	指针变量	158

6.3	指针运算	163
6.4	指针与数组	167
6.4.1	指针与一维数组	167
6.4.2	指针与二维数组	170
6.5	引用变量	171
6.6	动态存储分配	173
	习题	176
第7章	函数	182
7.1	函数的定义	182
7.1.1	定义格式	182
7.1.2	定义格式举例	182
7.1.3	有关函数定义的几点说明	183
7.2	函数的调用	186
7.2.1	调用格式	186
7.2.2	调用过程	187
7.2.3	函数调用举例	189
7.3	变量的作用域	194
7.3.1	作用域分类	194
7.3.2	程序举例	195
7.4	递归函数	202
7.5	函数重载	205
*7.6	模板函数	207
*7.7	函数指针	211
	习题	213
第8章	结构与联合	224
8.1	结构的定义	224
8.1.1	结构定义格式	225
8.1.2	定义格式举例	225
8.1.3	结构使用说明	226
8.2	结构变量的定义和初始化	227
8.2.1	用结构类型名定义变量	227
8.2.2	定义结构类型的同时定义变量	229
8.2.3	定义无名结构类型的同时定义变量	230
8.3	结构成员的访问操作	230
8.4	使用结构的程序举例	231
8.5	结构与函数	239

* 8.6	结构与链表	243
* 8.7	结构与操作符重载	246
8.8	联合	255
8.8.1	联合的定义和访问	255
8.8.2	使用联合举例	257
	习题	262
第 9 章	类与对象	266
9.1	类的定义	266
9.1.1	类的定义格式	266
9.1.2	定义格式举例	267
9.1.3	有关说明	269
9.2	构造函数	274
9.2.1	无参构造函数和带参构造函数	274
9.2.2	拷贝构造函数	280
9.2.3	赋值重载函数	281
9.2.4	构造函数中的初始化表	282
9.3	析构函数	283
9.4	友元函数和友元类	285
	习题	292
第 10 章	类的继承与多态性	301
10.1	类的继承	301
10.1.1	派生类定义的格式	301
10.1.2	格式举例	303
10.1.3	应用举例	311
* 10.2	类的虚函数与多态性	314
* 10.3	类的静态成员	319
* 10.4	模板类	321
	习题	323
第 11 章	C++ 流	329
11.1	C++ 流的概念	329
11.2	输入输出格式控制	331
11.2.1	ios 类中的枚举常量	331
11.2.2	ios 类中的成员函数	333
11.2.3	格式控制操作符	336
11.3	文件操作	338

11.3.1	文件的概念	338
11.3.2	文件的打开和关闭	340
11.3.3	字符文件的访问操作	342
11.3.4	字节文件的访问操作	349
* 11.4	字符串流	356
	习题	358
附录 A	ASCII 代码表	364

第 1 章 程序设计引论

1.1 数 制

数制就是记数的规则和制度。日常生活中使用最多的是十进制记数制,在计算机系统中采用的是二进制记数制。二进制记数制很方便在计算机中表示和进行处理,但它不符合人们记数的习惯,而且占用的位数较多,难于记忆和书写。所以人们往往向计算机输入十进制数据,这时它要被转换成对应的二进制数后才能保存和运算;当从计算机内向外部设备输出数据时,也需要做相反的转换,即把二进制数转换成对应的十进制数,使人们容易阅读和使用。有时也把二进制数转换成对应的八进制或十六进制数输出,因为二进制同它们之间的转换极其快捷和简便。

1.1.1 不同数制的表示

十进制记数采用 0、1、2、3、4、5、6、7、8、9 共 10 个记数符号和逢十向高位进一的规则,当然还可以使用正、负号和小数点等符号;二进制记数采用 0 和 1 这两个记数符号和逢二向高位进一的规则;八进制采用 0、1、2、3、4、5、6、7 共 8 个记数符号和逢八向高位进一的规则;十六进制采用 0~9 共 10 个数字和 A、B、C、D、E、F 6 个字母(大、小写等效)共 16 个记数符号和逢 16 向高位进 1 的规则,其中 A、B、C、D、E、F 分别对应十进制的 10、11、12、13、14 和 15。

在每一种记数制中,所使用的记数符号的个数称为该记数制的基数。由此可知,十进制的基数为 10,二、八、十六进制的基数分别为 2、8 和 16。

对于每一种记数制中的每个数据,从小数点向左第 i 位的权定义为基数的 $i-1$ 次方,从小数点向右第 i 位的权定义为基数的 $-i$ 次方。例如,对于十进制数,从小数点向左每一位的权依次为 10^0 、 10^1 、 10^2 、 \dots ,从小数点向右每一位的权依次为 10^{-1} 、 10^{-2} 、 10^{-3} 、 \dots ;对于二进制数,从小数点向左每一位的权依次为 2^0 、 2^1 、 2^2 、 \dots ,从小数点向右每一位的权依次为 2^{-1} 、 2^{-2} 、 2^{-3} 、 \dots ;同样,对于八进制数,从小数点向左每一位的权依次为 8^0 、 8^1 、 8^2 、 \dots ,从小数点向右每一位的权依次为 8^{-1} 、 8^{-2} 、 8^{-3} 、 \dots ;对于十六进制数,从小数点向左每一位的权依次为 16^0 、 16^1 、 16^2 、 \dots ,从小数点向右每一位的权依次为 16^{-1} 、 16^{-2} 、 16^{-3} 、 \dots 。

每一种记数制中一个数的大小,通常用对应的十进制数值来衡量,它等于该数中每一位数字与对应权值的乘积的累加和。假定一个记数制的基数为 k ,具有 n 个整数位和 m 个小数位的一个数 x 为 $S_{n-1}S_{n-2}\dots S_1S_0.S_{-1}S_{-2}\dots S_{-m}$,则 x 的大小为 $\sum_{i=-m}^{n-1} (S_i \times k^i)$,此累加式又称为该数的按权展开式。例如:

$$\begin{aligned}
 (376.45)_{10} &= \sum_{i=-2}^2 (S_i \times 10^i) \\
 &= 3 \times 10^2 + 7 \times 10^1 + 6 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} \\
 &= 376.45
 \end{aligned}$$

$$\begin{aligned}
 (10110.101)_2 &= \sum_{i=-3}^4 (S_i \times 2^i) \\
 &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\
 &= 16 + 0 + 4 + 2 + 0.5 + 0 + 0.125 \\
 &= 22.625
 \end{aligned}$$

$$\begin{aligned}
 (526.4)_8 &= \sum_{i=-1}^2 (S_i \times 8^i) \\
 &= 5 \times 8^2 + 2 \times 8^1 + 6 \times 8^0 + 4 \times 8^{-1} \\
 &= 320 + 16 + 6 + 0.5 = 342.5
 \end{aligned}$$

$$\begin{aligned}
 (2B0D)_{16} &= \sum_{i=0}^3 (S_i \times 16^i) \\
 &= 2 \times 16^3 + 11 \times 16^2 + 0 \times 16^1 + 13 \times 16^0 \\
 &= 8192 + 2816 + 13 \\
 &= 11021
 \end{aligned}$$

由于二进制记数制只需要使用 0 和 1 这两种记数符号,也就是说只需要用一种事物的两种不同的状态来对应表示,一种状态表示 0,另一种状态表示 1,这在现实世界中很容易实现。如可以利用二极管的导通和截止这两种不同状态来对应表示二进制的 0 和 1,用一块磁性区域的正向磁化和反向磁化来分别表示二进制的 0 和 1 等。在计算机系统中,正是利用电信号的高和低、有和无、正和反等两种状态来表示、存储和传送二进制数中的每一位数字信息的。

表 1.1 给出了十进制整数 0~20 所对应的二进制、八进制和十六进制数。

表 1.1 20 以内各种数制对照表

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0	0	0	11	1011	13	B
1	1	1	1	12	1100	14	C
2	10	2	2	13	1101	15	D
3	11	3	3	14	1110	16	E
4	100	4	4	15	1111	17	F
5	101	5	5	16	10000	20	10
6	110	6	6	17	10001	21	11
7	111	7	7	18	10010	22	12
8	1000	10	8	19	10011	23	13
9	1001	11	9	20	10100	24	14
10	1010	12	A				

1.1.2 不同数制的转换

每一种记数制中的每个数都可以通过按权展开式得到对应的十进制数。如二进制数 101101 所对应的十进制数为 $1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 = 45$; 八进制数 2473 所对应的十进制数为 $2 \times 8^3 + 4 \times 8^2 + 7 \times 8^1 + 3 \times 8^0 = 1339$; 十六进制数 AE2.4 所对应的十六进制数为 $10 \times 16^2 + 14 \times 16^1 + 2 \times 16^0 + 4 \times 16^{-1} = 2786.25$ 。

十进制整数转换成其他任一记数制可采用逐次除以相应记数制的基数 k 取余法, 直到整数商为 0 时止。转换过程具体表述为: 第一次用被转换的十进制整数除以相应记数制的基数 k 后所得整余数为相应记数制整数的最低位 S_0 , 第二次用第一次得到的整数商除以 k 后所得整余数为相应记数制整数的次最低位 S_1 , 依次类推, 假定共得到 n 个整余数, 最后一次整数商为 0 所得整余数为相应记数制整数的最高位 S_{n-1} 。例如, 若把十进制整数 185 分别转换成二进制、八进制和十六进制整数, 则计算过程分别如图 1.1(a)、(b) 和 (c) 所示。

$ \begin{array}{r} 2 \overline{)185} \quad \text{余数} \\ \underline{2 \ 92} \quad \dots 1 = S_0 \\ 2 \overline{)46} \quad \dots 0 = S_1 \\ \underline{2 \ 23} \quad \dots 0 = S_2 \\ 2 \overline{)11} \quad \dots 1 = S_3 \\ \underline{2 \ 5} \quad \dots 1 = S_4 \\ \underline{2 \ 2} \quad \dots 1 = S_5 \\ \underline{2 \ 1} \quad \dots 0 = S_6 \\ 0 \quad \dots 1 = S_7 \end{array} $ <p style="text-align: center;">(a)</p>	$ \begin{array}{r} 8 \overline{)185} \quad \text{余数} \\ \underline{8 \ 23} \quad \dots 1 = S_0 \\ 8 \overline{)23} \quad \dots 7 = S_1 \\ 0 \quad \dots 2 = S_2 \end{array} $ <p style="text-align: center;">(b)</p>	$ \begin{array}{r} 16 \overline{)185} \quad \text{余数} \\ \underline{16 \ 11} \quad \dots 9 = S_0 \\ 0 \quad \dots 11 = S_1 \end{array} $ <p style="text-align: center;">(c)</p>
---	---	---

图 1.1 十进制整数转换为其他进制数的过程

由图 1.1 的转换过程可知, 十进制整数 185 对应的二进制数为 10111001, 对应的八进制数为 271, 对应的十六进制数为 B9。

十进制纯小数转换成其他记数制的纯小数采用逐次乘以相应记数制的基数 k 取整法, 直到积的小数部分为 0 或达到所规定的精度为止。转换过程具体表述为: 第一次用被转换的十进制纯小数乘以相应记数制的基数 k 后所得乘积的整数部分是相应记数制纯小数的最高位 S_{-1} , 第二次用第一次乘积的小数部分乘以基数 k 后所得乘积的整数部分是相应记数制纯小数的次最高位 S_{-2} , 依次类推, 直到所得乘积的小数部分为 0 或已经求出所规定的小数位数为止。例如, 若把十进制纯小数 0.6845 分别转换成对应的二进制、八进制和十六进制纯小数, 则转换过程如图 1.2(a)、(b) 和 (c) 所示。

由图 1.2 可知, 十进制数 0.6845 对应的二进制数为 0.10101111, 假定只需保留二进制的 8 位小数; 0.6845 对应的八进制数为 0.53635, 假定只需保留八进制的 5 位小数; 0.6845 对应的十六进制数为 0.AF3B, 假定只需保留十六进制的 4 位小数。

若一个十进制数为混合小数, 则需将整数部分和小数部分分别转换成相应的记数制数, 然后把它们合并起来。例如, 若把十进制数 54.3 转换成二进制数, 则整数部分 54 转换成的二进制整数为 110110, 小数部分 0.3 转换成的二进制纯小数为 0.010011 (假定保