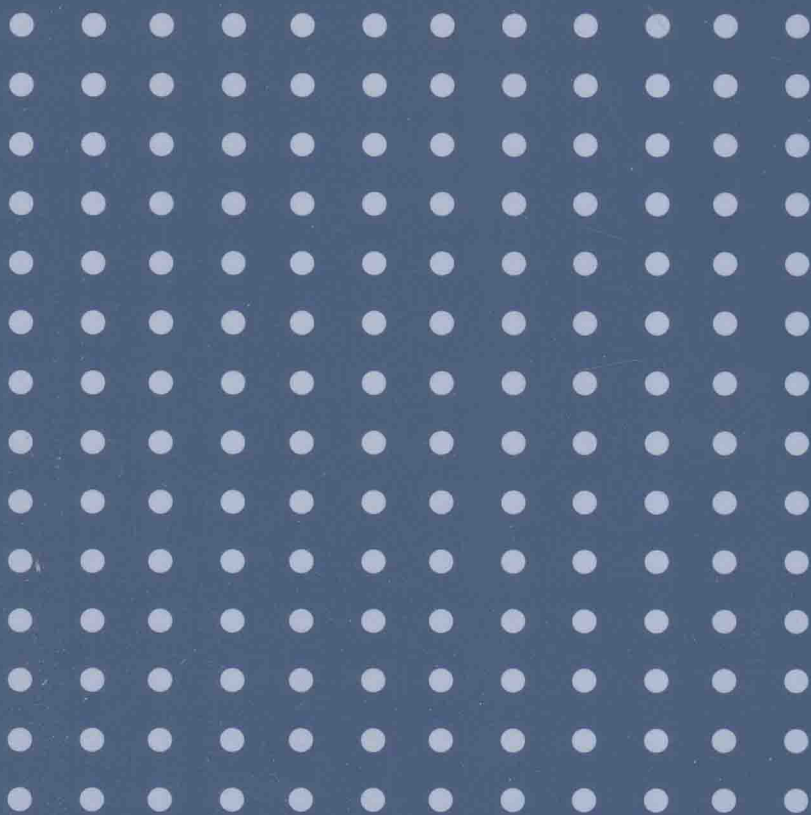


重点大学计算机专业系列教材

C++程序设计基础

揣锦华 编著

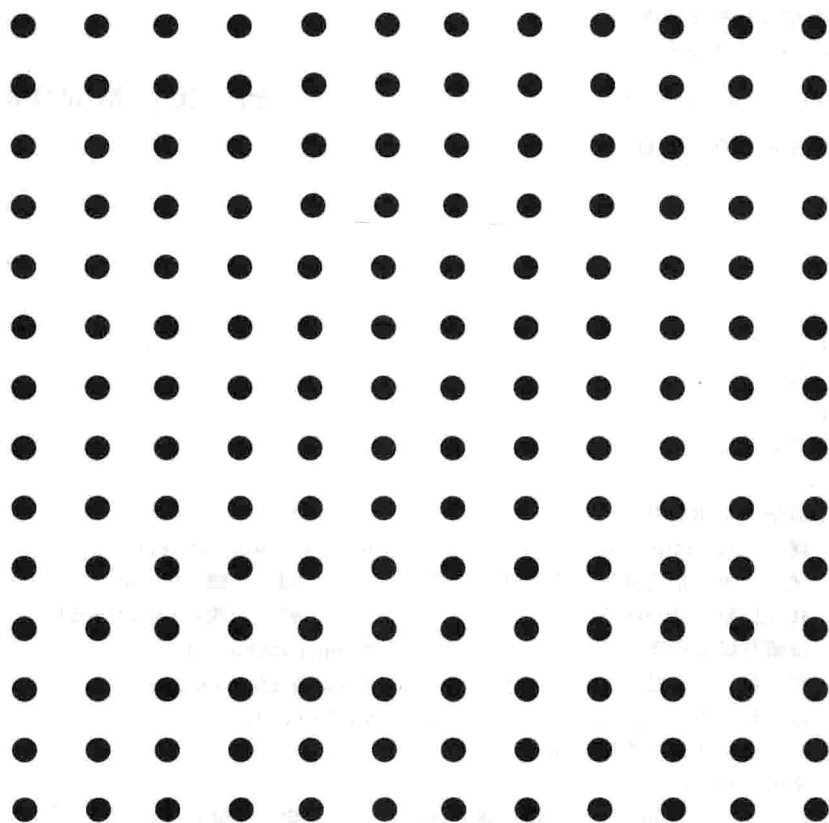


清华大学出版社

重点大学计算机专业系列教材

C++程序设计基础

揣锦华 编著



清华大学出版社

内 容 简 介

本书基于新的 C++ 标准,从程序设计基础知识开始,系统地介绍 C++ 语言的基本概念、语法规则和编程技术。本书是作者结合多年的 C 和 C++ 语言的教学实践经验编写而成,对每一部分的知识点和难点,都力求以比较精练的语言进行讲解,并按照由浅入深、循序渐进、前后贯穿的原则,精选了大量例题。各章在介绍 C++ 程序设计语言的基础上,采用发散性思维方法,对相关知识进行了扩展;同时,每章配备有相应的教学案例和大量习题,既方便教师安排教学,又便于读者综合运用所学知识,进一步提高编程技能。

本书可作为高等院校计算机专业或非计算机专业的第一门程序设计语言教学用书,也可作为程序设计人员的自学参考用书。

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++ 程序设计基础/揣锦华编著. —北京:清华大学出版社,2015

重点大学计算机专业系列教材

ISBN 978-7-302-38355-0

I. ①C… II. ①揣… 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(2014)第 243365 号

责任编辑:郑寅堃 薛 阳

封面设计:傅瑞学

责任校对:白 蕾

责任印制:刘海龙

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:三河市少明印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:18.25

字 数:443 千字

版 次:2015 年 2 月第 1 版

印 次:2015 年 2 月第 1 次印刷

印 数:1~1500

定 价:37.00 元

产品编号:059376-01

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大,社会对计算机专业人才的需求不仅体现在数量的增加上,而且体现在质量要求的提高上,培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前,我国共有 16 个国家重点学科、20 个博士点一级学科、28 个博士点二级学科集中在教育部属重点大学,这些高校在计算机教学和科研方面具有一定优势,并且大多以国际著名大学计算机教育为参照系,具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系,形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础,其中专业教材的使用和建设则是这种氛围的重要组成部分,一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势,特别是专业教材建设上的优势,同时配合各重点大学的计算机学科建设和专业课程教学需要,在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下,清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”,同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1. 面向学科发展的前沿,适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础,反映基本理论和原理的综合应用,重视实践和应用环节。
2. 反映教学需要,促进教学发展。教材要能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。规划教材建设的重点依然是专业基础课和专业主干课;特别注意选择并安排了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重点大学

计算机专业教学内容和课程体系改革成果的教材。

4. 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多种具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;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,择优落实。在制订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、评审确定主编。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。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,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。

教材编委会

前言

随着信息技术和计算机科学的发展,程序设计已成为现代人应该掌握的基本技能。C++语言是从C语言发展演变而来的一种面向对象的程序设计语言,C++全面兼容了C,同时提供了比C更严格更安全的语法,从这个意义上讲,C++首先是一个更好的C。虽然C++语言是从C语言发展而来的,但是C++本身也是一个完整的程序设计语言,因此,将C++作为程序设计的入门语言进行学习是完全可行的。

十多年前,本人曾出版过《C++程序设计语言》一书,近几年在具体的教学实践中,对C++程序设计的教学内容和形式有了一些新的认识和改进的想法。另外,随着C++的发展,原书内容和使用的标准也需要升级更新。2003年出版的书已使用多年,在社会上有一定的影响,广大读者对原书也提出了一些诚恳的改进意见,所以重新出版此书也实现了作者回馈热心读者的愿望。

本书基于新的C++标准,从程序设计基础知识开始,系统地介绍C++语言的基本概念、语法规则和编程技术。针对初学者的特点,力求做到深入浅出,将复杂的概念用简洁浅显的语言来讲述,使读者可以轻松地入门,循序渐进地提高。本书是作者总结多年教学实践经验编写而成,对每一部分的知识点和难点,都力求以比较精练的语言进行讲解,在介绍语法时,着重从程序设计方法学的角度讲述其意义和用途。同时,对重要的知识点都列举了必要的例题进行说明,并对例题所采用的算法和编程技术进行了深刻的分析,旨在使读者对C++编程技术不仅知其然,并知其所以然。在介绍基础知识的基础上,还选择了一些综合性较强的案例,对常用的数据结构和经典的算法进行详细的剖析,这些案例既方便教师安排教学,又便于读者综合运用所学知识,进一步提高编程技能。本书每章后面都安排有知识扩展部分,以引导读者进一步深入了解C和C++语言。另外,书中各章均配有大量的思考题和习题,方便读者练习和自我考核。

本书集众家所长,采用“传统教学+案例教学”方式,以满足不同层次读者的需求,并运用发散性思维方法,对相关知识进行扩展,意在开阔视野,培养编程兴趣,使读者在循序渐进中提高编程能力。

全书以程序设计方法贯穿始终,从语法规则到程序设计实践,力求在掌握基本程序设计方法的同时,培养读者良好的程序设计习惯,为今后的学习打下坚实的编程基础。本书的宗旨是不仅要使读者掌握 C++ 语言本身,而且能够对现实世界中的问题及其解决方法用 C++ 语言进行描述。学习程序设计的关键是要深刻领会程序设计的内涵,广学多练,培养对程序设计语言的“语感”,最终掌握程序设计的“秘笈”。

本书既可以作为高等学校计算机程序设计相关课程的教材和教学参考书,也可以作为学习程序设计人员的培训或自学教材。

本书的第 1 章至第 9 章由长安大学的揣锦华编写,第 10 章由长安大学的袁琪编写。在本书的编写过程中,编者查阅和参考了大量文献,在此对书后所列出的参考文献的作者一并表示感谢。另外,对广大读者和师生对本书诚恳的建议和意见也表示衷心的感谢。由于作者水平有限,书中难免存在不足和错误之处,恳请读者批评指正。

编 者

2014 年 10 月

目 录

第 1 章 程序设计基础知识	1
1.1 计算机中数的表示与编码	1
1.1.1 进位记数制	2
1.1.2 二进位数在计算机内的表示	4
1.1.3 常用的信息编码	6
1.2 程序设计及程序设计语言	7
1.3 算法及算法表示	8
1.4 用程序流程图表示算法	9
1.5 程序设计方法	11
1.5.1 结构化程序设计	11
1.5.2 面向对象程序设计	12
1.6 小结与知识扩展	12
1.6.1 小结	12
1.6.2 计算机中常用存储单位及其换算	13
习题	14
第 2 章 C++ 简单程序设计	15
2.1 C++ 程序的基本结构和要素	15
2.1.1 一个简单的 C++ 程序	15
2.1.2 字符集	17
2.1.3 词法记号	18
2.1.4 注释	19
2.2 基本数据类型和数据	19
2.2.1 基本数据类型	20
2.2.2 常量	22
2.2.3 变量和引用	25
2.3 运算符与表达式	27

2.3.1	算术表达式	28
2.3.2	自增和自减运算符	28
2.3.3	赋值表达式	30
2.3.4	逗号表达式	30
2.3.5	关系表示式	31
2.3.6	逻辑表达式	32
2.3.7	条件表达式	32
2.3.8	sizeof 运算符	33
2.3.9	位运算	33
2.3.10	运算符的优先级和结合性	36
2.4	数据类型转换	36
2.4.1	赋值时的类型转换	36
2.4.2	表达式中隐含转换	38
2.4.3	强制类型转换	38
2.5	简单的输入与输出控制	39
2.5.1	C++ 的输入与输出	39
2.5.2	通过 I/O 流控制符进行格式控制	40
2.6	C++ 基础知识编程案例	41
2.7	小结与知识扩展	43
2.7.1	小结	43
2.7.2	C 语言的 printf() 和 scanf() 函数	43
2.7.3	C 语言的 getchar() 和 putchar() 函数	45
2.7.4	数据溢出	46
习题		47
第 3 章	程序控制结构	50
3.1	选择控制结构	51
3.1.1	选择控制语句 if else	51
3.1.2	条件运算符?: 代替 if else 语句	52
3.1.3	if else 语句的嵌套	53
3.1.4	多路选择控制语句 switch	56
3.2	循环控制结构	57
3.2.1	while 语句	57
3.2.2	do while 语句	59
3.2.3	for 语句	60
3.2.4	输入信息控制循环	61
3.2.5	循环嵌套	62
3.3	其他控制语句	64
3.3.1	break 语句	64

3.3.2	continue 语句	64
3.3.3	goto 语句	65
3.4	程序控制编程案例	66
3.5	小结与知识扩展	73
3.5.1	小结	73
3.5.2	字符函数库	74
	习题	76
第 4 章	函数	80
4.1	函数的定义与调用	80
4.1.1	函数定义	81
4.1.2	函数调用	82
4.2	函数的参数传递	84
4.2.1	数值传递	85
4.2.2	引用传递	86
4.3	递归函数	87
4.4	函数探幽	89
4.4.1	默认参数值的函数	89
4.4.2	内联函数	91
4.4.3	函数重载	93
4.4.4	函数模板	95
4.5	使用 C++ 系统函数	97
4.6	函数编程案例	98
4.7	小结与知识扩展	102
4.7.1	小结	102
4.7.2	main() 函数	103
	习题	104
第 5 章	数组	109
5.1	数组的基本概念	109
5.2	一维数组	110
5.2.1	一维数组的声明	110
5.2.2	访问一维数组的元素	110
5.2.3	一维数组的初始化	112
5.3	多维数组	113
5.3.1	二维数组的声明	114
5.3.2	访问二维数组的元素	114
5.3.3	二维数组的初始化	115
5.4	数组作为函数参数	117

5.4.1	一维数组名作为参数	117
5.4.2	二维数组的行地址作为参数	120
5.5	数组与字符串	122
5.5.1	字符型数组的初始化	122
5.5.2	字符串的基本操作	123
5.6	数组编程案例	126
5.6.1	排序	126
5.6.2	查找	127
5.6.3	统计	129
5.6.4	字符处理	131
5.6.5	数列处理	132
5.7	小结与知识扩展	134
5.7.1	小结	134
5.7.2	数组越界	134
5.7.3	算法的时间复杂度及其表示	135
	习题	137
第 6 章	自定义数据类型	140
6.1	结构体	140
6.1.1	结构体类型的定义	140
6.1.2	结构体变量的引用	143
6.1.3	结构体数组	144
6.2	共用体	146
6.3	枚举类型	147
6.4	类型自定义语句	149
6.5	类和对象	150
6.5.1	类的定义	150
6.5.2	对象的使用	151
6.5.3	构造函数和析构函数	153
6.6	小结与知识扩展	154
6.6.1	小结	154
6.6.2	构造函数的重载	154
	习题	155
第 7 章	指针	156
7.1	指针的概念	156
7.2	指针变量	158
7.2.1	指针变量的声明	158
7.2.2	指针的基本操作	159

7.2.3 指针变量的初始化	159
7.2.4 指针的运算	161
7.3 指针与数组	164
7.3.1 指针与数组的关系	164
7.3.2 使用指针访问数组元素	165
7.3.3 指向多维数组的指针	166
7.3.4 多级指针	168
7.3.5 指针数组	170
7.4 指针与函数	171
7.4.1 指针变量作为函数参数	171
7.4.2 指向函数的指针	174
7.4.3 指针作为函数的返回类型	176
7.5 指针与字符串	177
7.5.1 字符型指针与字符串	177
7.5.2 字符串标准库函数	179
7.6 动态内存分配与 new 和 delete 运算符	180
7.6.1 new 运算符	180
7.6.2 delete 运算符	182
7.7 指针编程案例	183
7.8 小结与知识扩展	190
7.8.1 小结	190
7.8.2 malloc() 和 free() 函数	191
7.8.3 常指针	193
7.8.4 链表操作	194
习题	200
第 8 章 C++ 程序结构	202
8.1 作用域	202
8.2 生存期	205
8.3 局部变量和全局变量	206
8.4 编译预处理	207
8.4.1 文件包含	208
8.4.2 宏定义	208
8.4.3 条件编译	213
8.5 多文件结构	215
8.6 名称空间	216
8.7 小结与知识扩展	219

8.7.1 小结	219
8.7.2 命令行参数	220
8.7.3 异常处理	222
习题	225
第9章 输入/输出流与文件操作	226
9.1 输入/输出流概述	226
9.1.1 输入/输出流的概念	226
9.1.2 输入/输出标准流	227
9.2 控制输出格式	228
9.2.1 使用控制符	228
9.2.2 使用流对象 cout 的成员函数	229
9.3 使用成员函数精确控制输入/输出	230
9.3.1 输入函数	230
9.3.2 输出函数	233
9.4 串流类	233
9.5 文件流类	234
9.5.1 文件的概念	234
9.5.2 文件的读写操作	235
9.6 小结与知识扩展	237
9.6.1 小结	237
9.6.2 C 语言的文件操作函数	238
习题	243
第10章 VC++调试器	246
10.1 建立应用程序调试版本	247
10.1.1 调试版本与发行版本	247
10.1.2 项目调试创建	247
10.1.3 设置开发环境参数	251
10.2 基于 IDE 的调试	256
10.2.1 使用调试窗口	257
10.2.2 断点调试	257
10.2.3 运行调试器	263
10.2.4 调试器窗口	263
10.2.5 使用调试器	263
10.3 高级调试技巧	268
10.3.1 调试过程中异常处理的设置	268

10.3.2	异常处理编程	269
10.4	小结与知识扩展	273
10.4.1	小结	273
10.4.2	调试线程	273
习题		274
参考文献		276

程序设计基础知识

第 1 章

本章要点

- 计算机中数的表示与编码
- 程序设计概念
- 用程序流程图表示算法

随着信息技术的迅猛发展,计算机程序设计的运用领域越来越广泛,程序设计语言也层出不穷。那么,什么是程序设计?有哪些程序设计语言?如何进行程序设计呢?这些都是计算机程序设计初学者首先遇到的问题。

无论什么计算机程序设计语言,其程序设计的思想和方法都相差无几,关键是要深刻领会程序设计的内涵,广学多练,培养自己对程序设计语言的“语感”,逐渐掌握程序设计的“秘笈”,最终让程序设计成就你的精彩人生。

作为全书的开篇,本章从计算机中数的表示与编码开始,对程序设计的概念和所涉及的基本知识进行概括性的介绍,旨在使读者对程序设计有一个初步的认识,为后面章节的学习储备一些基础知识。下面就让我们从充满魔力的二进制数 0 和 1 开始,进入计算机程序设计的魔幻世界。

1.1 计算机中数的表示与编码

计算机最主要的功能是处理信息,如处理数值、文字、声音、图形和图像等,在计算机内部,各种信息都必须经过数字化编码后才能被传送、存储和处理。因此,掌握信息编码的概念与处理技术是至关重要的。

所谓编码,就是采用少量的基本符号,选用一定的组合原则,表示大量复杂多样的信息。信息编码的两大要素是基本符号的种类和这些符号的组合规则。例如,用 10 个阿拉伯数码表示数字,用 26 个英文字母表示英文词汇等,都是编码的典型例子。

在计算机中,广泛采用的是只用 0 和 1 两个基本符号组成的基 2 码,或称为二进制码。在计算机中采用二进制码的原因如下。

(1) 二进制码在物理上最容易实现。例如,可以只用高、低两个电平表示 0 和 1,也可以用脉冲的有无或者脉冲的正负极性表示它们。

(2) 二进制码用来表示的二进制数,其编码、计数、加减运算比较简单。

(3) 二进制码的两个符号 0 和 1 正好与逻辑命题的两个值“是”和“否”,或“真”和“假”相对应,为计算机实现逻辑运算和程序中的逻辑判断提供了便利的条件。

1.1.1 进位记数制

在日常生活中,最常用的进位计数制是我们非常熟悉的十进制。在计算机程序设计中,除十进制外,还有二进制、八进制和十六进制等。

1. 进位计数制的表示

在采用进位计数的数字系统中,如果只用 r 个基本符号(例如 $0, 1, 2, \dots, r-1$)表示数值,则称其为基 r 数制(Radix- r Number System), r 称为该数制的基(Radix)。如常用的十进制(Decimal)数,就是 $r=10$,即基本符号为 $0, 1, 2, \dots, 9$ 。如取 $r=2$,即基本符号为 0 和 1,则为二进制(Binary)数。

对于不同的数制,它们的共同特点如下。

(1) 每一种数制都有固定的符号集(Symbol Set):如十进制数制,其符号有 10 个,分别是 $0, 1, 2, \dots, 9$;二进制数制,其符号有两个,即 0 和 1。

(2) 都使用位置表示法(Notation):即处于不同位置的数符所代表的值不同,其值与它所在位置的权值有关。

例如,十进制数 555.55 可表示为:

$$555.55 = 5 \times 10^2 + 5 \times 10^1 + 5 \times 10^0 + 5 \times 10^{-1} + 5 \times 10^{-2}$$

可以看出,各种进位计数制中权的值恰好是基数的某次幂(Power)。因此,对任何一种进位计数制表示的数都可以写出按其权展开的多项式之和。任意一个 r 进制数 N 可表示为:

$$N = \sum_{i=m-1}^{-k} D_i \times r^i \quad (1-1)$$

式中的 D_i 为该数制采用的基本数符, r^i 是权, r 是基数, m 是整数部分的最大位数, k 是小数部分的最大位数。

2. 不同进制数之间的转换

(1) r 进制数转换为十进制数

式(1-1)本身就提供了将 r 进制数转换为十进制数的方法。

例如,把二进制数 11010.101 转换成相应的十进制数。

$$\begin{aligned} (11010.101)_B &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= (26.625)_D \end{aligned}$$

(2) 十进制数转换为 r 进制数

这里,整数部分和小数部分的转换方法是不相同的,下面分别介绍。

① 整数部分的转换

把一个十进制的整数不断除以所需要的基数 r ,取其余数(即除 r 取余法),就能够转换

成以 r 为基数的数。

例如,为了把十进制的整数转换成相应的二进制数,只要把十进制数不断除以 2,并记下每次所得余数(余数总是 1 或 0),所有余数连起来即为相应的二进制数,这种方法称为除 2 取余法。

例如,把十进制数 25 转换成二进制数。

2	25	余数
2	12	1 最底位
2	6	0
2	3	0
2	1	1
	0	1 最高位

所以有: $(25)_D = (11001)_B$

注意:第一位余数是低位,最后一位余数是高位。

② 小数部分转换

要将一个十进制小数转换成 r 进制小数时,可将十进制小数不断地乘以 r ,并取整,这称为乘 r 取整法。

例如,将十进制数 0.625 转换成相应的 r 进制数。

	0.625	取整
×	2	
	1.250	1 最高位
×	2	
	0.500	0
×	2	
	1.000	1 最低位

所以, $(0.625)_D = (0.101)_B$

如果十进制数包含整数和小数两部分,则必须先将十进制小数点两边的整数和小数部分分开,分别完成相应转换,然后,再把 r 进制整数和小数部分组合在一起。

例如,将十进制数 25.625 转换成 r 进制数,只要将上例整数和小数部分组合在一起即可,即 $(25.625)_D = (11001.101)_B$ 。

3. 非十进制数间的转换

通常两个非十进制数之间的转换方法是采用上述两种方法的组合,即先将转换数转换为相应的十进制数,然后再将十进制数转换为其他进制数。至于二进制、八进制和十六进制数之间的转换,由于它们之间存在特殊关系,即 $8^1 = 2^3$, $16^1 = 2^4$, 因此,它们之间的转换方法比较容易,如表 1-1 所示。

表 1-1 十进制、二进制、八进制和十六进制之间的关系

十进制数 (Decimal)	二进制数 (Binary)	八进制数 (Octonary)	十六进制数 (Hexadecimal)
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3