



高等学校“十二五”规划教材·计算机软件工程系列

Programming Designing Fundamentals (The C language)

程序设计基础

(C 语言)

© 董宇欣 主编



注重编程基本功培养/经典实例引入程序设计

突破传统语言主线/贯穿结构化程序设计思想



哈尔滨工业大学出版社
HARBIN INSTITUTE OF TECHNOLOGY PRESS

高等学校“十二五”规划教材·计算机软件工程系列

程序设计基础(C语言)

董宇欣 主编

哈尔滨工业大学出版社

内容简介

本书从培养应用型人才的角度出发,系统地介绍了 C 语言编程的基本知识和程序设计的基本方法,内容包括:程序设计基础、简单的程序设计、基本数据类型、运算符和表达式、控制结构、数组、函数、指针、字符串、编译预处理、数据构造类型和多文件、文件。各章配有大量例题和练习。

本书可以作为面向应用的高等院校中计算机类各个专业程序设计基础课程、一般高等院校理工科各专业公共计算机基础课程“高级语言程序设计”、“程序设计基础”、“C 程序设计”、“C 语言”等的教材和参考书,还可以供从事计算机工作的有关人员参考。

图书在版编目(CIP)数据

程序设计基础:C 语言/董宇欣主编. —哈尔滨:
哈尔滨工业大学出版社,2011.9
高等学校“十二五”规划教材·计算机软件工程系列
ISBN 978-7-5603-3392-2

I. ①程… II. ①董… III. ①C 语言-程序设计
IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 181522 号

策划编辑 王桂芝 赵文斌
责任编辑 刘 瑶 唐 蕾
出版发行 哈尔滨工业大学出版社
社 址 哈尔滨市南岗区复华四道街 10 号 邮编 150006
传 真 0451-86414749
网 址 <http://hitpress.hit.edu.cn>
印 刷 哈尔滨市工大节能印刷厂
开 本 787mm×1092mm 1/16 印张 17 字数 424 千字
版 次 2011 年 9 月第 1 版 2011 年 9 月第 1 次印刷
书 号 ISBN 978-7-5603-3392-2
定 价 34.00 元

(如因印装质量问题影响阅读,我社负责调换)

高等学校“十二五”规划教材

计 算 机 软 件 工 程 系 列

编 审 委 员 会

名誉主任 丁哲学

主 任 王义和

副 主 任 王建华

编 委 (按姓氏笔画排序)

王霓虹	印桂生	许少华	任向民
衣治安	刘胜辉	苏中滨	张 伟
苏建民	李金宝	苏晓东	张淑丽
沈维政	金 英	胡 文	姜守旭
贾宗福	黄虎杰	董宇欣	

随着计算机工程的发展和社会对计算机工程人才需求的增长,软件工程专业培养目标更加明确,特色更加突出。目前,国内多数高校软件工程专业培养目标是以需求为导向,注重培养学生掌握软件工程基本理论、专业知识和基本技能,具备运用先进的工程化方法、技术和工具从事软件系统分析、设计、开发、维护和管理等工作能力,以及具备参与工程项目的实践能力、团队协作能力、技术创新能力和市场开拓能力,具有发展成软件行业高层次工程技术和企业管理人才潜力的,适应社会主义市场经济和信息产业发展需要的“工程实用型”人才。

本系列教材针对软件工程专业“突出学生的软件开发能力和软件工程素质,培养从事软件项目开发和管理的工程技术人员”的培养目标,集9家软件学院(软件工程专业)的优秀作者和强势课程,本着“立足基础,注重实践应用;科学统筹,突出创新特色”的原则,精心策划编写。具体特色如下:

1. 紧密结合企业需求,多校优秀作者联合编写

本系列教材编写在充分进行企业需求、学生需要、教师授课方便等多方市场调研的基础上,采取了校企适度联合编写的做法,根据目前企业的普遍需要,结合在校学生的实际学习情况,校企作者共同研讨、确定课程的安排和相关教材内容,力求使学生在校学习过程中就能熟悉和掌握科学研究及工程实践中需要的理论知识和实践技能,以便适应就业及创业的需要,满足国家对软件工程人才的需要。

2. 多门课程系统规划,注重培养学生工程素质

本系列教材精心策划,从计算机基础课程→软件工程基础与主干课程→设计与实践课程,系统规划,统一编写。既考虑到每门课程的相对独立性,基础知识的完整性,又兼顾相关课程之间的横向联系,避免知识点的简单重复,力求形成科学、完整的知识体系。

本系列教材中的《离散数学》、《数据库系统原理》、《算法设计与分析》等基础教材在引入概念和理论时,尽量使其贴近社会现实及软件工程等学科的技术和应用,力图将基本知识与软件工程学科的实际问题结合起来,在具备直观性的同时强调启发性,必须让学生理解所

学的知识。《软件工程导论》、《软件体系结构》、《软件质量保证与测试》、《软件项目管理》等软件工程主干课程以《软件工程导论》为线索,各课程间相辅相成,互相照应,系统地介绍了软件工程的整个学习过程。《数据结构应用设计》、《编译原理设计与实践》、《操作系统设计与实践》、《数据库系统设计与实践》等实践类教材以实验为主题,坚持理论内容以必须和够用为度,实验内容以新颖、实用为原则编写。通过一系列实验,培养学生的探究、分析问题的能力,激发学生的学习兴趣,充分调动学生的非智力因素,提高学生的实践能力。

相信本系列教材的出版,对于培养软件工程人才、推动我国计算机软件工程事业的发展必将起到积极作用。

A handwritten signature in black ink, appearing to read '王义中' (Wang Yizhong), written in a cursive style.

2011 年 7 月

◎ 前言

Preface

程序设计是高等学校计算机专业和软件工程专业的专业基础课程,目的是向学生介绍程序设计的基础知识,使学生掌握高级语言程序设计的基本思想和方法,培养学生掌握用计算机处理问题的思维方法,为后续专业课的学习打下良好的基础。

本书以 C 语言为工具,以程序设计为主线,采用任务驱动式、案例式教学组织内容。书中采用任务驱动方式提出问题,使学生逐步形成用程序设计语言去描述问题和解决问题的能力;书中每个知识点都以典型案例为基础引入并剖析知识点,注重工程性、实践性和应用性。全书贯穿了提出问题、分析问题、引出概念、讲解知识点、程序实现的编写思路,突破传统的单纯以计算机语言为主线编排内容,旨在由浅入深、循序渐进地培养学生的程序设计思维习惯和方法。

本书自始至终贯穿结构化程序设计思想,所有例题都具有良好的结构和程序设计风格,并且每个知识点后都提出程序设计中容易出现的问题,使读者一开始编写程序就具有良好的编程风格,而且不容易犯常见的错误。本书不但适合教学使用,而且适合学生自学。

本书共分 12 章,第 1 章引言,介绍了程序设计的相关概念;第 2 章数据类型、运算符与表达式,介绍了 C 语言的基本语言元素;第 3 章顺序结构程序设计,介绍了 C 语言的基本语句及输入输出方法;第 4 章选择结构程序设计,介绍了使用选择结构解决问题的方法;第 5 章循环结构程序设计,介绍了循环结构程序设计的方法;第 6 章数组,介绍了使用数组存储批量数据及数组的处理方法;第 7 章函数,介绍了函数的定义、调用过程;第 8 章指针,介绍了通过指针访问数据的方法及指针操作;第 9 章结构体,介绍了构造数据类型结构的知识及链表的概念;第 10 章程序的结构,介绍了变量的作用域和存储类型及动态内存分配的方法;第 11 章再论函数,进一步讨论了函数的调用方法及指针与函数;第 12 章文件,介绍了文件的概念及基本操作方法。

本书由哈尔滨工程大学董宇欣和吴良杰、哈尔滨师范大学于延、哈尔滨学院潘莹、黑龙江工程学院苗志滨共同编写,其中董宇欣任主编,潘莹、苗志滨、于延任副主编,吴良杰、毕武统稿并主审。本书具体编写分工如下:董宇欣负责第 1~3 章,潘莹负责第 4、5、12 章,吴良杰、于延负责第 6、8、9 章,苗志滨负责第 7、10、11 章。

由于作者水平有限,书中难免会有错误或疏漏之处,真诚地欢迎各位专家和读者批评指正,以帮助我们进一步完善教材。

编 者

2011 年 7 月

◎ 目 录

Contents

第1章 引言	1
1.1 程序设计基础	1
1.1.1 程序设计的概念	1
1.1.2 程序设计语言的发展	1
1.1.3 程序设计思想	3
1.2 C语言的特点及发展过程	4
1.2.1 C语言的产生	4
1.2.2 C语言的特点	4
1.2.3 C语言的发展过程	5
1.3 程序与算法	6
1.3.1 程序	6
1.3.2 算法简介	6
1.3.3 算法的表示	7
1.4 C程序开发过程	12
1.4.1 程序实例	12
1.4.2 程序的开发过程	13
1.4.3 Turbo C++ 3.0 集成开发环境	14
习题	16
第2章 数据类型、运算符与表达式	18
2.1 数据类型	18
2.1.1 引入数据类型的原因	18
2.1.2 基本数据类型	18
2.1.3 字符集	19
2.1.4 关键字和标识符	20
2.2 常量	20
2.2.1 整型常量	21
2.2.2 实型常量	22
2.2.3 字符型常量	23
2.2.4 字符串常量	24

2.2.5	符号常量	24
2.3	变量	25
2.3.1	变量的定义及初始化	25
2.3.2	定义变量时应注意的问题	26
2.4	运算符与表达式	26
2.4.1	运算符及其优先级与结合性	27
2.4.2	算术运算符	29
2.4.3	关系运算符	29
2.4.4	逻辑运算符	30
2.4.5	赋值运算符	32
2.4.6	++、--运算符	33
2.4.7	逗号运算符	35
2.4.8	条件运算符	36
2.5	类型转换	36
2.5.1	自动转换	36
2.5.2	强制类型转换	37
	习题	38
第3章	顺序结构程序设计	40
3.1	C语言的语句	40
3.2	赋值语句	41
3.3	基本输入输出	42
3.3.1	格式输入与输出	42
3.3.2	字符数据的输入与输出	49
3.4	顺序结构程序实例	51
	习题	55
第4章	选择结构程序设计	58
4.1	选择结构	58
4.2	选择结构的实现方法	58
4.2.1	单分支	58
4.2.2	双分支	59
4.2.3	多分支	62
4.2.4	选择结构的嵌套	67
4.3	选择结构的分析与使用时的问题	68
	习题	68
第5章	循环结构程序设计	72
5.1	循环结构	72

5.2 循环结构实现方法	72
5.2.1 for 循环语句	72
5.2.2 while 循环	75
5.2.3 do-while 循环	77
5.2.4 嵌套的循环	78
5.2.5 几种循环结构的比较及使用中注意的问题	80
5.3 流程的转移语句	81
5.3.1 break 语句	81
5.3.2 continue 语句	81
5.3.3 应用程序举例	82
5.4 常用算法	85
习题	87
第6章 数组	90
6.1 数组的引入	90
6.2 数组的定义与使用	91
6.2.1 数组的定义及初始化	92
6.2.2 数组的使用	94
6.2.3 字符数组	97
6.2.4 程序举例	104
6.3 常用算法	108
6.3.1 排序	108
6.3.2 查找	111
6.3.3 数制转换	112
习题	114
第7章 函数	118
7.1 定义函数的原因	118
7.2 函数的定义	120
7.2.1 无参函数的定义	120
7.2.2 有参函数的定义	120
7.3 函数的执行过程	121
7.3.1 函数的调用	121
7.3.2 函数的原型声明	123
7.3.3 函数的参数	124
7.3.4 函数的返回值	124
7.4 数组与函数	127

7.4.1 一维数组与函数传递	127
7.4.2 二维数组与函数的传递	130
7.5 程序举例	131
习题	136
第8章 指针	138
8.1 指针的概念	138
8.1.1 地址和内存的访问方式	139
8.1.2 指针和指针变量	140
8.2 指针的定义与使用	140
8.2.1 指针的定义与初始化	140
8.2.2 指针操作	141
8.3 指针做函数参数	143
8.4 指针和数组	144
8.4.1 指针的运算	144
8.4.2 一维数组的地址与指针	145
8.4.3 二维数组的地址与指针	150
8.4.4 程序举例	152
8.5 指针与字符串	156
8.6 指针数组和指针的指针	161
8.6.1 指针数组	161
8.6.2 指针的指针	164
习题	165
第9章 结构体	169
9.1 结构体类型与结构体变量	169
9.1.1 结构体类型的声明	169
9.1.2 结构体变量的定义及初始化	171
9.1.3 结构体变量的使用	173
9.2 结构体数组	175
9.2.1 结构体数组的定义及初始化	175
9.2.2 结构体数组应用	177
9.3 结构体指针	179
9.4 结构体与函数	182
9.5 指针处理链表	184
9.5.1 链表	184
9.5.2 链表的建立	184

9.5.3 链表的插入	186
9.5.4 链表的删除	187
习题	188
第 10 章 程序的结构	192
10.1 编译预处理	192
10.1.1 宏定义	192
10.1.2 文件包含	197
10.1.3 条件编译	199
10.2 变量的作用域和存储类型	202
10.2.1 变量的作用域	202
10.2.2 变量的存储类别	206
10.3 动态内存分配	210
10.3.1 动态内存分配函数	211
10.3.2 动态数组的实现	212
10.4 多函数、多文件程序的编写	214
习题	218
第 11 章 再论函数	221
11.1 函数的嵌套调用	221
11.2 函数的递归调用	222
11.3 指针与函数	225
11.3.1 返回指针值的函数	225
11.3.2 函数的指针和指向函数的指针变量	227
11.4 程序举例	229
习题	232
第 12 章 文件	234
12.1 文件的概述	234
12.1.1 流	234
12.1.2 存储设备的使用	234
12.1.3 目录及文件格式	235
12.2 文件的打开与关闭	236
12.2.1 文件的打开	236
12.2.2 文件的关闭	238
12.3 文件的读写	239
12.3.1 文件的读操作	239
12.3.2 文件的写操作	244

12.4 文件的其他常用函数	249
12.4.1 文件的定位	249
12.4.2 文件的随机读写	250
12.4.3 其他相关函数	251
习题	251
附录	255
附录 A 常用字符与 ASCII 代码对照表	255
附录 B C 库函数	256
参考文献	258

第 1 章

引 言

1.1 程序设计基础

1.1.1 程序设计的概念

在计算机得到广泛应用的今天,无论是科学技术领域,还是工作和娱乐,计算机都给我们带来很多的便利。我们可以利用计算机制图、制表、设计 3D 模型、聊天、玩游戏、看电影,等等。我们可以在计算机上完成各种各样的任务,计算机已经成为人们生活和工作中不可或缺的一部分。计算机之所以会拥有如此强大的功能,除了计算机本身提供硬件支持,更重要的原因就是人们开发了无数的。能够指挥计算机完成各种任务的程序,而程序设计就是用某种计算机语言编写这些程序的过程。《维基百科全书》给出了比较详细的程序设计的定义:程序设计是给出解决特定问题的程序的过程,是软件构造活动中的重要组成部分。

程序设计的最终目的就是用程序来控制计算机为人们解决特定的问题。“软件构造活动中的重要组成部分”,说明程序设计在软件构造中的角色,软件最终需要通过程序的运行来发挥作用。程序设计是将问题转化成引导计算机运行的指令序列,并在计算机中运行得到正确结果的过程,需要借助程序设计语言来完成设计。

程序设计一般包括以下几个步骤:

(1)分析待解决的问题并建立相应的数学模型。分析问题,找出已知和未知,确定解决问题的步骤,然后将解题过程转化成数学表达式,并建立各种变量及常量之间的关系,从而建立起解决问题的数学模型。数学模型的建立十分关键,它决定了程序的正确性和复杂程度。

(2)确定数据结构和算法。根据建立的数学模型,确定存放数据的数据结构,针对所确定的数据组织形式选择合适的算法。

(3)编程。根据确定的数据结构及算法,使用选定的计算机语言编写程序代码,输入计算机中。

(4)调试和运行程序。消除程序的语法或逻辑错误,用各种可能的输入数据对程序进行测试,分析结果,对不合理的数据进行适当的处理,直至运行获得预期的结果。

(5)整理并写出文档资料。

1.1.2 程序设计语言的发展

由于机器只能识别由 0、1 组成的二进制代码,所以最早使用的计算机语言是指令和数

据均由二进制代码组成的机器语言。随后发明的汇编语言可以将机器指令映射为一些能被人读懂的助记符,如 ADD、SUB 等。运行汇编程序时,先将用助记符描述的源程序转换成机器指令程序,然后通过运行机器指令程序得到输出结果。随着计算机技术的发展,FORTRAN、BASIC、PASCAL、C、C++和 Java 等高级语言应运而生,高级语言的出现使计算机程序设计不再过度地依赖某种特定的机器或环境。

程序设计语言主要经历了机器语言、汇编语言和高级语言三个发展阶段,图 1.1 直观地表示了程序设计语言的发展历程。

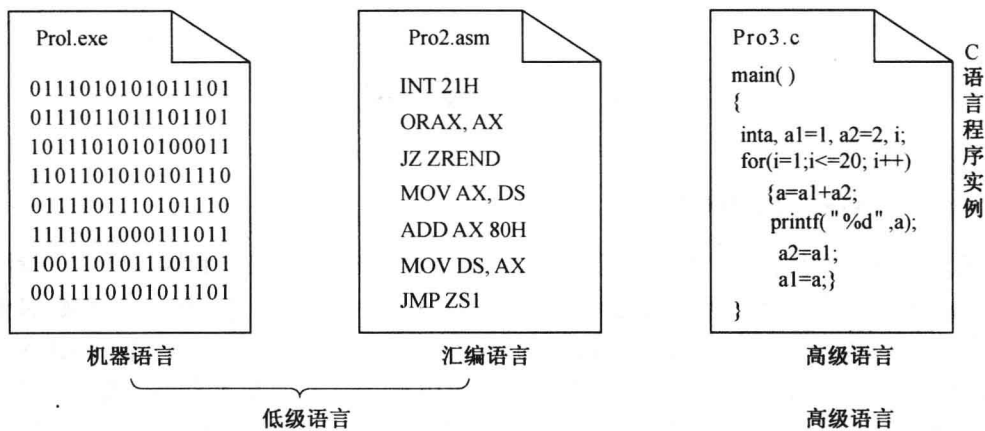


图 1.1 程序设计语言的发展

1. 机器语言

从 19 世纪起,随着机械式计算机的更新,出现了穿孔卡片,通过卡片上的孔洞引导计算机做相应的操作。世界上第一台电子数字计算机 ENIAC 使用的就是穿孔卡片,在卡片上使用的是专家们才能理解的语言,由于它与人类的自然语言之间存在巨大的鸿沟,人们称之为机器语言,也就是第一代计算机语言。机器语言是最底层的计算机语言,这种语言在本质上是计算机能识别的唯一语言。机器语言直接由机器指令(二进制)构成,因此,由它编写的计算机程序不需要转换就可直接被计算机系统识别并运行,执行速度快、效率高。但它也存在着严重的缺点:机器语言难掌握、编程繁琐、可读性差、易出错,并且依赖于具体的机器,通用性差。由于不同型号计算机的指令系统往往各不相同,所以在一种计算机上执行的程序要在另一种不同指令系统的计算机上执行,必须重新编写程序,造成了重复工作。

2. 汇编语言

汇编语言是第二代程序设计语言,为了克服机器语言难掌握、编程繁琐、可读性差和易出错的缺点,人们采用能帮助记忆的英文缩写符号(称为指令助记符)来代替机器语言指令代码中的操作码,用地址符号来代替地址码。用指令助记符及地址符号书写的指令称为汇编指令(也称符号指令),而用汇编指令编写的程序称为汇编语言源程序。汇编语言又称符号语言。计算机不能直接识别和理解用符号语言编写的程序,需要一个专门的系统程序负责将汇编语言程序翻译成机器语言程序,这种翻译程序就是汇编程序。任何类型的计算机都配有只适用于自己的汇编程序。尽管汇编语言与机器语言相比有不少的优势,但缺点仍然很明显。由于这种“助记符”语言基本上是与机器指令一一对应的,汇编语言与机器语言

的差别仅仅体现在表示形式上,所以汇编语言和机器语言一样,依赖于具体的计算机型号,用汇编语言写的程序不能方便地移植到另一种机器上。汇编语言同样存在机器语言的缺点,如依赖计算机硬件、缺乏通用性、编程繁琐、易出错等。汇编语言程序的特点迫使程序员必须从机器语言角度去思考,编写代码仍然不方便。因此,汇编语言和机器语言统称为低级语言。汇编语言编写的程序相对于高级语言编写的程序,其目标程序占用内存空间少,运行速度快,在某些场合发挥着高级语言不可替代的作用。

3. 高级语言

机器语言和汇编语言都是面向机器的语言,要求编程人员必须对机器硬件结构及工作原理十分熟悉,这对非计算机专业人士来说是很困难的。高级语言是第三代计算机程序设计语言,它接近人类的自然语言和数学公式,同时又不依赖于具体的硬件,编写出的程序能在不同的机器上运行。1957年,第一个完全脱离机器硬件的高级语言——FORTRAN(FORmula TRANslator)在IBM公司研发成功。随后的50多年来,共有几百种高级语言出现,使用较普遍的有FORTRAN、COBOL(Common Business Oriented Language)、BASIC、LISP、Pascal、C、C++、VC(Visual C++)、VB(Visual BASIC)、Delphi、C#、JAVA等。高级语言又分为面向过程的语言和面向对象语言。

高级语言容易学习和掌握,程序可读性好,可维护性强,可靠性高。高级语言与具体的计算机硬件关系不大,程序可移植性好,重用率高。用高级语言编写的程序要比低级语言容易,同时程序的编制和调试过程大大简化,编程效率得到大幅度的提高。

1.1.3 程序设计思想

随着计算机语言的不断发展,程序设计思想也在不断地发生变化,最初的程序只能针对特定类型的计算机进行设计,之后出现了面向过程语言,模块化设计和自顶向下、逐步细化的结构化编程方法提高了编码效率,面向对象编程技术的出现使编程更接近人类思维习惯。

1. 结构化程序设计思想

结构化程序设计的概念是E. W. Dijkstra在20世纪60年代末提出的,其实质是控制编程中的复杂性。他的主要观点是采用自顶向下、逐步求精的程序设计方法,使用三种基本控制结构构造程序。结构化程序设计强调程序设计风格和程序结构的规范化,提倡清晰的程序结构,把一个复杂问题的求解过程分阶段进行,每个阶段处理的问题都控制在人们容易理解和处理的范围内。

具体来说,采用以下方法确保得到结构化的程序:

- (1) 自顶向下。
- (2) 逐步求精。
- (3) 模块化设计。
- (4) 结构化编码。

如图1.2所示,自顶向下、逐步细化的过程是将问题求解由抽象逐步具体化的过程。具体在程序设计中,根据程序模块的功能将一个复杂的问题划分为若干个模块,每一个子模块完成一项独立的功能,每个模块内部只采用顺序、选择(分支)和循环三种规范化的控制结构来构造。

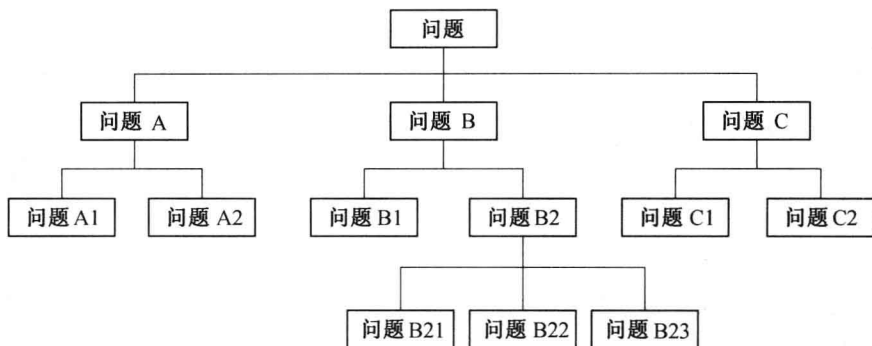


图 1.2 自顶向下、逐步细化的结构化程序设计思想

按这种思想编写出来的结构化程序具有结构清晰、容易理解、容易验证、便于开发和维护等特点。

2. 面向对象程序设计思想

随着面向对象语言的出现,面向对象程序设计方法也应运而生并得到迅速发展。面向对象程序设计是面向过程设计方法的继承和发展,是基于一种自然、朴素的思想来模拟人类思维习惯的程序设计方法,它具有抽象、封装、继承和多态性等特征。客观世界是由各种各样的对象(或称实体、事物)组成的,每个对象都有自己的内部状态和运动规律,不同对象间的相互联系和相互作用构成了各种不同的系统,进而构成整个客观世界。面向对象软件开发过程就是使用计算机语言将现实世界映射到计算机世界的过程。面向对象程序设计在软件开发领域掀起了巨大的变革,极大地提高了软件开发效率。

1.2 C 语言的特点及发展过程

1.2.1 C 语言的产生

C 语言是当今世界最为流行的计算机高级语言之一,它是由美国的 Dennis Ritchie 于 1972 年设计的,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。

C 语言由早期的编程语言 BCPL (Basic Combind Programming Language) 发展演变而来。1970 年,美国电话电报公司(AT&T)贝尔实验室的 Ken Thompson 根据 BCPL 语言设计了较先进的 B 语言, C 语言在 B 语言基础上增加了数据类型的概念,以及其他功能强大的特性。1978 年,贝尔实验室正式发表了 C 语言。

1.2.2 C 语言的特点

C 语言之所以能够存在且持续地发展,是因为 C 语言本身存在许多不同于其他语言的特点。C 语言是比较流行的高级程序设计语言,许多大型软件,尤其是系统软件和商业软件,都是用 C 语言编写的。C 语言是一种健壮的语言,其丰富的内置函数集和运算符可用来编写任意复杂的程序。C 语言的主要特点如下:

(1) 语言简洁,结构紧凑,程序书写方便,使用灵活。C 语言只有 32 个关键字,9 种控制