



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言程序设计

The C Programming Language

王洪海 陈向阳 盛魁 邵立 主编

杜秀全 包得海 薛峰 李凌峰 文萍芳 徐丽萍 余会娟 副主编

- 内容可读性与实用性并重
- 知识讲授与能力培养并重
- 例题分析与习题训练并重



高校系列

 人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化普通高等教育“十二五”规划教材立项项目

C语言程序设计

The C Programming Language

王洪海 陈向阳 盛魁 邵立 主编

杜秀全 包得海 薛峰 李凌峰 文萍芳 徐丽萍 余会娟 副主编

赵守忠 郑诚 主审



高校系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C语言程序设计 / 王洪海等主编. -- 北京 : 人民邮电出版社, 2011. 7
21世纪高等学校计算机规划教材. 高校系列
ISBN 978-7-115-25307-1

I. ①C… II. ①王… 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 (2011) 第080481号

内 容 提 要

本书共分 10 章, 首先简单介绍程序设计基础, 然后分别系统地介绍 C 语言的数据类型与运算符、C 语言的 3 种程序结构、数组与函数、指针与文件等相关知识。

本书面向初学者, 语言叙述通俗易懂, 概念清晰, 实践性强。本书提供了大量的实例与习题, 注重各部分知识的综合应用训练。

本书适合作为高等学校本科、高职高专、成人高校和其他初学者学习 C 程序设计的教材, 也可供参加全国计算机等级考试 (二级 C) 的读者选用。

工业和信息化部普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

C 语言程序设计

-
- ◆ 主 编 王洪海 陈向阳 盛 魁 邵 立
副 主 编 杜秀全 包得海 薛 峰 李凌峰 文萍芳
徐丽萍 余会娟
主 审 赵守忠 郑 诚
责任编辑 邹文波
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
中国铁道出版社印刷厂印刷
- ◆ 开本: 787×1092 1/16
印张: 18 2011 年 7 月第 1 版
字数: 384 千字 2011 年 7 月北京第 1 次印刷

ISBN 978-7-115-25307-1

定价: 35.00 元

读者服务热线: (010) 67170985 印装质量热线: (010) 67129223
反盗版热线: (010) 67171154

参编及合作单位

(排名不分先后)

安徽三联学院

中国科学技术大学

安徽大学

甘肃民族师范学院

解放军电子工程学院

安庆医药高等专科学校

亳州职业技术学院

池州职业技术学院

安徽医学高等专科学校

前 言

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。在对操作系统、系统应用及需要对硬件进行操作的场合中,用 C 语言明显优于其他高级语言,因此,许多大型应用软件都是用 C 语言编写的。由于 C 语言功能强、使用灵活、可移植性好、目标程序质量好,从而受到编程者广泛的欢迎。

本书和同类书相比,注重可读性和实用性,难点分散,努力用人们易于理解的方式清楚地叙述复杂的概念,具有体系合理、逻辑清楚、例题丰富、通俗易懂的特点。本书在编写过程中,遵循知识讲授和能力训练并重的原则,在讲清基本知识的基础上,注意了例题的选择,增加了例题和习题的数量。讲述中力求理论联系实际和循序渐进,注重培养读者分析问题和程序设计的能力,使读者养成良好的程序设计风格 and 习惯。

全书共分 10 章,内容包括程序设计基础,C 语言概述,数据类型、运算符及表达式,程序控制结构,数组,函数,指针,结构体与共用体,文件,编译预处理。其中,第 1 章、第 3 章由王洪海编写,第 2 章由陈向阳编写,第 4 章由徐丽萍编写,第 5 章、第 9 章由包得海编写,第 6 章由杜秀全、薛峰编写,第 7 章由盛魁编写,第 8 章由邵立编写,第 10 章由李凌峰、余会娟、文萍芳编写。全书由王洪海统稿并定稿。本书由中国科学技术大学教授、现安徽三联学院信息与通信技术系主任赵守忠教授和安徽大学计算机科学与技术学院副院长郑诚教授主审,安徽三联学院院长领导对于本书的出版也给予了大力的支持,在此一并表示衷心的感谢。

本书适合作为高等学校本科、高职高专、成人高校和其他初学者学习 C 程序设计的教材,也可供参加全国计算机等级考试(二级 C)的读者选用。为了方便读者学习,本书配有实验指导教材《C 语言程序设计实验指导》,帮助读者梳理所学知识,掌握应会和必会内容。

由于编者水平有限,书中难免存在错误和不妥之处,望广大读者提出宝贵意见,邮件请发至 sanlian_whh@163.com。

编者

2011 年 5 月

目 录

第 1 章 程序设计基础.....1

- 1.1 程序设计的基本概念.....1
 - 1.1.1 程序.....1
 - 1.1.2 算法与数据结构.....2
- 1.2 程序设计的一般步骤.....6
- 1.3 结构化程序设计方法.....7
- 小结.....7
- 习题 1.....8

第 2 章 C 语言概述.....9

- 2.1 C 语言的发展及其特点.....9
 - 2.1.1 C 语言的发展简史.....9
 - 2.1.2 C 语言的特点.....9
- 2.2 简单的 C 程序介绍.....10
 - 2.2.1 简单的 C 程序.....10
 - 2.2.2 C 源程序的结构特点.....12
 - 2.2.3 C 源程序的书写规则.....12
- 2.3 运行 C 语言程序的基本步骤.....13
- 小结.....13
- 习题 2.....14

第 3 章 数据类型、运算符及表达式.....15

- 3.1 C 语言的数据类型.....15
- 3.2 标识符与关键字.....16
- 3.3 常量.....16
 - 3.3.1 整型常量.....17
 - 3.3.2 实型常量.....18
 - 3.3.3 字符型常量.....18
 - 3.3.4 字符串常量.....20
 - 3.3.5 符号常量.....22
- 3.4 变量.....23
 - 3.4.1 变量的定义.....23
 - 3.4.2 变量的初始化.....25
- 3.5 运算符及其表达式.....28

- 3.5.1 算术运算符和算术表达式.....29
- 3.5.2 赋值运算符和赋值表达式.....32
- 3.5.3 关系运算符和关系表达式.....34
- 3.5.4 逻辑运算符和逻辑表达式.....35
- 3.5.5 条件运算符和条件表达式.....38
- 3.5.6 逗号运算符和逗号表达式.....39
- 3.5.7 位运算.....40
- 3.5.8 指针运算符、sizeof 运算符.....43

- 3.6 运算符的优先级和结合性.....43
- 3.7 表达式中数据类型的转换.....44
- 小结.....49
- 习题 3.....50

第 4 章 结构化的 C 语言程序设计.....54

- 4.1 结构化程序设计概述.....54
- 4.2 顺序结构程序设计.....54
 - 4.2.1 赋值语句、复合语句.....54
 - 4.2.2 字符输出函数.....55
 - 4.2.3 字符输入函数.....55
 - 4.2.4 格式化输出函数.....56
 - 4.2.5 格式化输入函数.....58
 - 4.2.6 顺序结构程序设计举例.....60
- 4.3 选择结构程序设计.....62
 - 4.3.1 简单的 if 结构.....62
 - 4.3.2 if-else 语句结构.....63
 - 4.3.3 if 结构的嵌套.....64
 - 4.3.4 if-else-if 语句结构.....65
 - 4.3.5 switch 结构.....67
 - 4.3.6 选择结构程序设计举例.....68
- 4.4 循环结构程序设计.....69
 - 4.4.1 goto 语句.....69
 - 4.4.2 while 语句.....69
 - 4.4.3 do-while 语句.....70
 - 4.4.4 for 语句.....71
 - 4.4.5 3 种循环语句的比较.....73
 - 4.4.6 循环嵌套.....73

4.4.7 break 语句	74	习题 6	148
4.4.8 continue 语句	74	第 7 章 指针	153
4.4.9 空语句	75	7.1 指针与指针变量的概念	153
4.4.10 循环结构程序设计举例	75	7.1.1 指针的概念	153
小结	77	7.1.2 指针变量	153
习题 4	78	7.2 指针变量的定义和引用	155
第 5 章 数组	83	7.2.1 指针变量的定义	155
5.1 一维数组的定义和引用	83	7.2.2 指针变量的赋值	156
5.1.1 一维数组的定义	83	7.2.3 指针变量的引用	156
5.1.2 一维数组元素的引用	85	7.3 指针和地址运算	159
5.1.3 一维数组的初始化	86	7.4 指针与数组	161
5.1.4 一维数组程序举例	87	7.4.1 指针变量与数组	161
5.2 二维数组	90	7.4.2 指针变量在一维数组中的应用	163
5.2.1 二维数组的定义	90	7.4.3 指针变量在多维数组中的应用	165
5.2.2 二维数组元素的引用	91	7.5 指针与字符串	169
5.2.3 二维数组的初始化	91	7.6 指针数组与多级指针	172
5.2.4 二维数组程序举例	92	7.6.1 指针数组的定义	172
5.3 字符数组	96	7.6.2 指针数组的使用	172
5.3.1 字符数组的定义	96	7.6.3 多级指针	174
5.3.2 字符数组的初始化	96	7.7 指针变量与函数	176
5.3.3 字符数组的引用	97	7.7.1 函数的操作方式与指针变量	176
5.3.4 字符串	97	7.7.2 指针型函数的定义与使用	177
5.3.5 字符串的输入/输出	98	7.7.3 函数指针的定义与使用	178
5.3.6 字符串操作函数	101	7.7.4 与指针有关的函数参数传递方式	179
5.3.7 字符数组应用举例	103	7.7.5 带参数的 main 函数和命令行参数	185
小结	106	7.8 指针与动态内存分配	186
习题 5	106	7.8.1 动态存储的概念	186
第 6 章 函数	111	7.8.2 C 语言的动态存储管理方式	186
6.1 函数概述	111	小结	187
6.1.1 函数的定义	113	习题 7	189
6.1.2 函数的调用	115	第 8 章 结构体与共用体	194
6.1.3 变量的生命期及作用域	120	8.1 结构体类型和结构体变量	194
6.2 函数参数传递	128	8.1.1 结构体类型的定义	194
6.2.1 值传递	130	8.1.2 结构体变量的定义	195
6.2.2 地址传递	132	8.1.3 结构体变量的引用	197
6.3 递归函数	139	8.1.4 结构体变量的初始化	198
6.4 内部函数与外部函数	145	8.2 结构体数组	198
小结	147	8.2.1 结构体数组的定义	198

8.2.2 结构体数组的引用	199	9.3.3 读/写字符串函数	229
8.2.3 结构体数组的初始化	200	9.3.4 读/写数据函数	231
8.3 结构体类型指针	201	9.3.5 格式读/写函数	233
8.3.1 指向结构体变量的指针	201	9.4 其他函数	234
8.3.2 指向结构体数组的指针	203	9.4.1 文件头定位函数	234
8.3.3 结构体变量和指向结构体指针 作为函数参数	203	9.4.2 文件随机定位函数	235
8.4 动态内存分配	205	9.4.3 出错检测函数	236
8.5 共用体	207	小结	236
8.5.1 共用体类型的定义	208	习题 9	237
8.5.2 共用体变量的定义	208	第 10 章 编译预处理	238
8.5.3 共用体变量的引用和初始化	209	10.1 宏定义	238
8.6 枚举类型	210	10.1.1 不带参数的宏定义	238
8.6.1 枚举类型的说明	210	10.1.2 带参数的宏定义	240
8.6.2 枚举型变量的定义	211	10.2 文件包含处理	242
8.7 用 typedef 定义类型	213	10.3 条件编译	245
小结	214	小结	247
习题 8	215	习题 10	248
第 9 章 文件	220	附录 A C 语言的字符集- ASC II 字符表	249
9.1 文件概述	220	附录 B C 语言的关键字	250
9.1.1 文件的概念	220	附录 C C 语言的库函数	251
9.1.2 文件类型指针	222	附录 D Turbo C 常见英文错误 提示及含义	259
9.2 文件的打开与关闭	223	附录 E 习题参考答案	263
9.2.1 文件的打开	223		
9.2.2 文件的关闭	225		
9.3 文件的读写	226		
9.3.1 文件尾测试函数	226		
9.3.2 读/写字符函数	227		

第 1 章

程序设计基础

1.1 程序设计的基本概念

电子计算机是 20 世纪人类最伟大、最杰出的科技发明之一，对人类社会的发展具有极其深远的影响。目前，计算机已经广泛而深入地渗透到人类社会的各个领域，极大地增加了人类认识世界和改造世界的能力。

一个完整的计算机系统由硬件系统和软件系统两大部分所组成。硬件是计算机系统物理装置的总称，它可以是电子的、机械的、光/电的元件或装置；软件是指在计算机硬件上运行的各种程序、数据和一些相关的文档、资料等。如果没有程序，计算机什么也不会做。

1.1.1 程序

从自然语言的角度来说，程序是对解决某个问题的方法步骤的描述；从计算机的角度来说，程序是用某种计算机能识别并可执行的计算机语言描述解决问题的方法步骤。

程序的特点是有始有终、每个步骤都能操作、所有步骤执行完对应问题要能得到解决。

例 1.1 求解任一圆的面积。

操作步骤如下。

第 1 步：输入圆的半径 R 。

第 2 步：利用公式 $S=3.14 \times R \times R$ ，求出圆的面积 S 。

第 3 步：输出结果 S 。

以上操作步骤就是求解任一圆面积的程序。

例 1.2 求 1~100 的和。

操作步骤如下。

第 1 步：初始化变量 $S=0$ 和计数器 $I=1$ 。

第 2 步： $S=S+I$

$I=I+1$

第 3 步：判断计数器 I 的值有没有大于 100，如果没有，返回到第二步执行；否则执行第 4 步

第 4 步：输出结果 S 的值。

以上操作步骤就是求 1~100 的和程序。

1.1.2 算法与数据结构

著名计算机科学家沃斯 (N.Wirth) 曾经提出过一个经典公式：程序=算法+数据结构。

1. 算法

程序设计的关键是解题的方法与步骤——算法，它反映了计算机的执行过程，是对解决特定问题操作步骤的一种描述。

算法可分为两大类：数值运算算法和非数值运算算法。数值运算算法：即求数值解，通过运算得出一个具体值。数值运算一般有现成的模型，算法较成熟。非数值运算算法：用于事务管理，如图书检索、人事管理等。具体对语言而言，其语法就是工具，是算法的一个具体实现和描述。所以，在程序设计的学习中，一方面要熟练掌握高级语言的语法，因为它是算法实现的基础，另一方面，还要掌握程序设计的基本方法，更重要的是必须认识算法的重要性，加强思维训练，达到写出高质量程序的目的。下面先看两个常用的简单算法。

例 1.3 累加求： $1+3+5+\cdots+99$ 。

操作步骤如下。

第 1 步： $1+3 \rightarrow S$ (结果)

第 2 步： $S+5 \rightarrow S$ (结果)

第 3 步： $S+7 \rightarrow S$ (结果)

⋮

第 50 步： $S+99 \rightarrow S$ (结果)

这样的算法虽然正确，但太烦琐。

改进的算法：

第 1 步：初始化变量 $S=0$ 和计数器 $I=1$ 。

第 2 步： $S=S+I$

$I=I+2$

第 3 步：判断计数器 I 的值有没有大于 99，如果没有，返回到第 2 步执行；否则执行第四步。

第 4 步：输出结果 S 的值。

一个优秀的算法应该具备以下特性。

(1) 有穷性：一个算法在合理的范围内应包含有限的操作步骤，而不能是无限的。

(2) 确定性：算法中每一个步骤应当是唯一的和确定无误的，而不能出现含糊、模棱两可而产生歧义的。

(3) 有零个或多个输入：在执行算法时，需从外界得到必要的信息。

(4) 有一个或多个输出：算法的目的是为了求解，解就是得到的输出。一个算法得到的结果就是该算法的输出，没有输出的算法是没有任何意义的。

(5) 有效性：算法中每一个步骤应当能有效地执行，并得到确定的结果。

对于程序设计人员，必须会设计算法，并根据算法写出程序。

2. 常用算法的表示方法

算法的表示方法有很多种，常用的有自然语言描述、伪代码、流程图、N-S图等。在这里重点介绍流程图和N-S图。

(1) 用自然语言表示算法。

用自然语言表示算法的优点是通俗易懂，但文字冗长，易产生歧义。除了很简单的问题，一般不用自然语言表示算法。

(2) 用流程图表示算法。

流程图是一种传统的算法表示法，它利用几何图形框来表示各种不同性质的操作，用流程线来指示算法的执行方向。

一个流程图包括：表示相应操作的框，带箭头的流程线，框内外必要的文字说明。几何图形框的含义如图1-1所示。



图 1-1 几何图形框的含义

例 1.4 用流程图表示例 1.3，表示如图 1-2 所示。

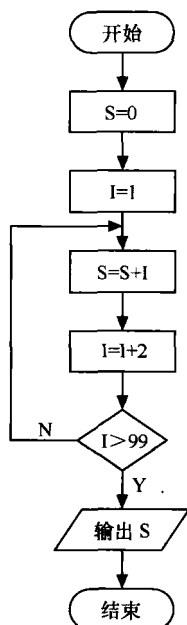


图 1-2 流程图示例

从上述例子可以看出，用流程图表示算法，直观形象，易于理解，是表示算法的良好工具。

注意，画流程线时，不要忘记画箭头，因为它是反应流程的执行先后次序的，如不画出箭头就很难判断各框的执行次序了。

根据结构化程序设计的思想，任何一个程序都由顺序、循环、选择 3 种基本结构组成，其流程图结构介绍如下。

① 顺序结构：虚线框内是一个顺序结构，其中 A 和 B 两个框是顺序执行的，即先执行 A，再执行 B。顺序结构是最简单的一种基本结构。

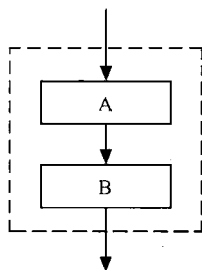


图 1-3 顺序结构

② 选择结构：选择结构又称分支结构。虚线框内是一个选择结构。注意，无论 P 条件是否成立，只能执行 A 框或 B 框之一，不可能两者都执行，如图 1-4 左图所示。其中，A 或 B 两个框中可以有一个是空的，即不执行任何操作，如图 1-4 右图所示。

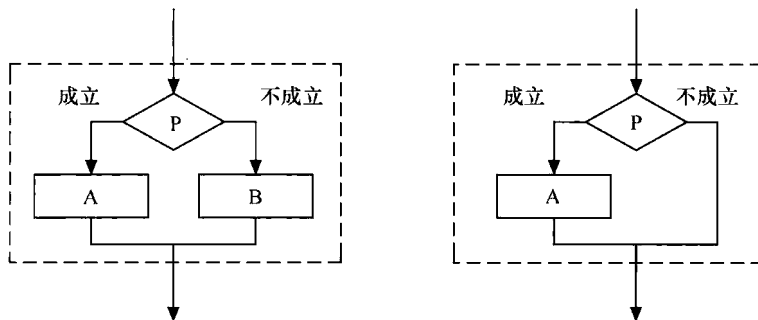


图 1-4 选择结构

③ 循环结构：循环结构又称重复结构，即反复执行某一部分的操作。循环结构有两种类型。

a. 当（while）型循环结构。当型循环结构如图 1-5 左图所示。它的功能是：当给定的条件 P 成立时，执行 A 框操作，执行完 A 后，再判断条件 P 是否成立，如果仍然成立，再执行 A 框，如此反复执行 A 框，直到某次 P 条件不成立为止。此时不再执行 A 框，而是脱离循环结构。

b. 直到（until）型循环结构。直到型循环结构如图 1-5 右图所示。它的功能是：先执行

A 框, 然后判断给定的条件 P 是否成立, 如果条件 P 不成立, 则再执行 A 框, 然后再对条件 P 作判断, 如条件 P 仍不成立, 又执行 A 框, 如此反复执行 A 框, 直到给定的条件 P 成立为止, 此时不再执行 A 框, 而是脱离循环结构。

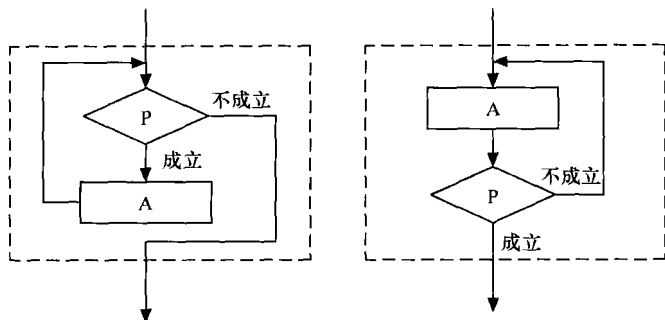


图 1-5 循环结构

以上 3 种基本结构, 具有以下共同特点:

- 只有一个入口;
- 只有一个出口;
- 结构内的每一部分都有机会被执行到;
- 结构内不存在“死循环”。

(3) 用 N-S 流程图表示算法。

用流程线表示算法的优点是直观形象、表示清晰、易于理解, 缺点是流程图占篇幅较多, 当算法复杂时, 画流程图费时且不方便。1973 年美国学者提出了一种新型流程图, 即 N-S 流程图。N-S 流程图适合于结构化程序设计, 因而作为编程人员, 传统流程图和 N-S 图都应掌握其使用方法。用 N-S 流程图表示顺序结构、选择结构和循环结构分别如图 1-6、图 1-7 和图 1-8 所示。

① 顺序结构:

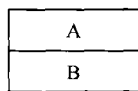


图 1-6 N-S 流程图顺序结构

② 选择结构:

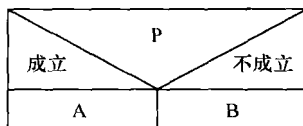


图 1-7 N-S 流程图选择结构

③ 循环结构:

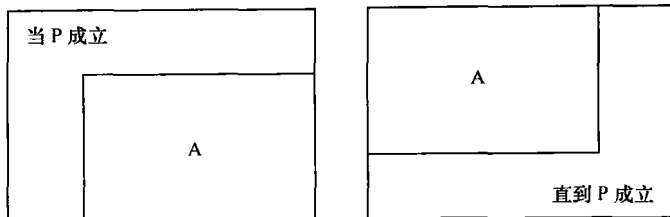


图 1-8 N-S 流程图循环结构

(4) 用伪代码表示算法。

伪代码就是介于自然语言和计算机语言之间的文字和符号来描述算法的，它不必遵守严格的语法规则。

(5) 用计算机语言表示算法。

我们的任务是用计算机解题，就是用计算机实现算法。用计算机语言表示算法必须严格遵守所用的语言的语法规则，这是和伪代码不同的。下面将对前面介绍过的例 1.4 的算法用 C 语言表示。

例 1.5 用 C 语言表示例 1.4。

```
(1_5.c)
#include<stdio.h>
void main()
{
    int i,s;
    i=2;
    s=0;
    do
    {
        s=s+i;
        i=i+2;
    }
    while(i<=99);
    printf("sum is %d\n",s);
}
```

数据结构是对参与运算的数据及它们之间的关系进行的描述，算法和数据结构是程序的两个重要方面。

1.2 程序设计的一般步骤

程序设计过程一般包括 3 个基本步骤。

第 1 步：分析问题。

作为一名程序设计者，首先要弄清楚用户的具体需求，在这里要进行用户需求分析（弄清楚用户要解决什么问题）、数据及处理分析（解决问题时需要哪些原始数据、怎么获得以及怎样处理这些数据等）、可行性分析（确定用户的需求在现有的条件下是否可解）、运行环

境分析（计算机的硬件和软件是否能满足设计的需求）等。

设计者在以上分析的基础上，将实际的问题抽象化并建立相应的数学模型，最终确定设计方案。

第 2 步：确定算法。

根据选取的数学模型和确定的设计方案，设计出可操作的具体步骤，并以流程图的形式将算法清晰、直观地表示出来。

第 3 步：编程实现。

程序设计的最后一步就是选择某种计算机语言编写源程序。对于源程序，应包含注释的语句，以描述程序各个部分做何种工作。此外，编程实现还应包含程序的调试与运行。调试主要是为了测试程序在运行过程中出现的一些错误。一旦程序运行情况良好以后，该源程序应以文件的形式固定保留下来，以便于维护和修改。

1.3 结构化程序设计方法

结构化程序设计是由迪克斯特拉（E.W.dijkstra）在 1969 年提出的，是以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，这样的每一个模块，不会受到其他模块的牵连，因而可将原来较为复杂的问题简化为一系列简单模块的设计，使完成每一个模块的工作变得单纯而明确，为设计一些较大的软件打下了良好的基础。结构化程序设计的具体方法为：

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

结构化程序设计的基本思想是采用“自顶向下，逐步求精”的程序设计方法和“单入口单出口”的控制结构。自顶向下、逐步求精的程序设计方法从问题本身开始，经过逐步细化，将解决问题的步骤分解为由基本程序结构模块组成的结构化程序框图；“单入口单出口”的思想认为一个复杂的程序，如果它仅是由顺序、选择和循环 3 种基本程序结构通过组合、嵌套构成，那么这个新构造的程序一定是一个单入口单出口的程序。据此就很容易编写出结构良好、易于调试的程序来。

小 结

本章简要介绍了程序设计的基本概念、什么是程序以及程序设计的一般步骤、结构化程

序设计方法等。本章的难点是给出一个问题，怎样设计出合适的算法并表示出来。

习 题 1

1. 什么是程序？
2. 程序设计过程一般有哪 3 个基本步骤？
3. 结构化程序设计的方法是什么？

第 2 章

C 语言概述

2.1 C 语言的发展及其特点

2.1.1 C 语言的发展简史

1972 年，贝尔实验室的布朗·W.卡尼汉和丹尼斯·M.利奇对 B 语言进行了完善和扩充，在保留 B 语言强大的硬件处理能力的基础上，扩充了数据类型，恢复了通用性，实现了最初的 C 语言。此后，为了让 C 语言成为在任何计算机上都能运行的通用计算机语言，1977 年，由两人合写了著名的“THE C PROGRAMMING LANGUAGE”一书。

随着微型机的普及，出现了版本不同的 C 语言，为了统一标准，美国国家标准协会（American National Standards Institute）于 1987 年制定了一个 C 语言标准，通常称之为 ANSI C。

C 语言是当今最流行的程序设计语言之一，它适合作为系统描述语言，既可以用来编写系统软件，也可以用来编写应用软件。目前最流行的 C 语言有以下几种：

- Microsoft C 或称 MS C；
- Borland Turbo C 或称 Turbo C；
- AT&T C。

这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。

2.1.2 C 语言的特点

C 语言之所以发展迅速，而且成为最受欢迎的语言之一，主要是因为它具有强大的功能。许多著名的系统软件，如 UNIX/Linux、Windows、dBASE III PLUS、dBASE IV 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序，就更能显示 C 语言的优势，像 PC-DOS、