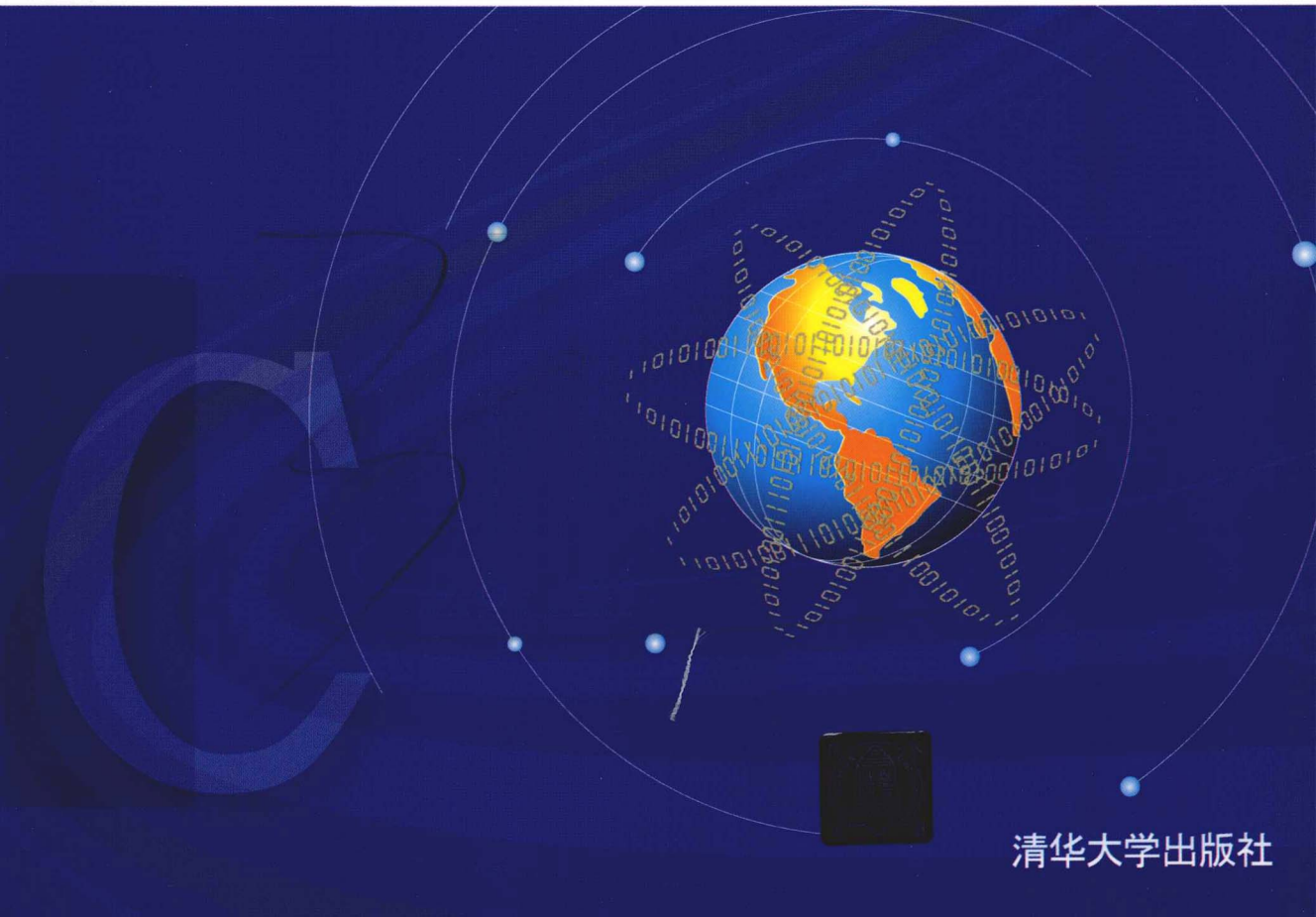
高等学校计算机类规划教材

- 以三层次逐级递进教学模式设计组织教材内容
- 注重解题思路、算法实现和程序设计思想
- 集教程和实验指导于一体，教程与实验指导有机结合
- 配套有PPT电子教案，例题、习题与实验的程序源代码

C语言程序设计

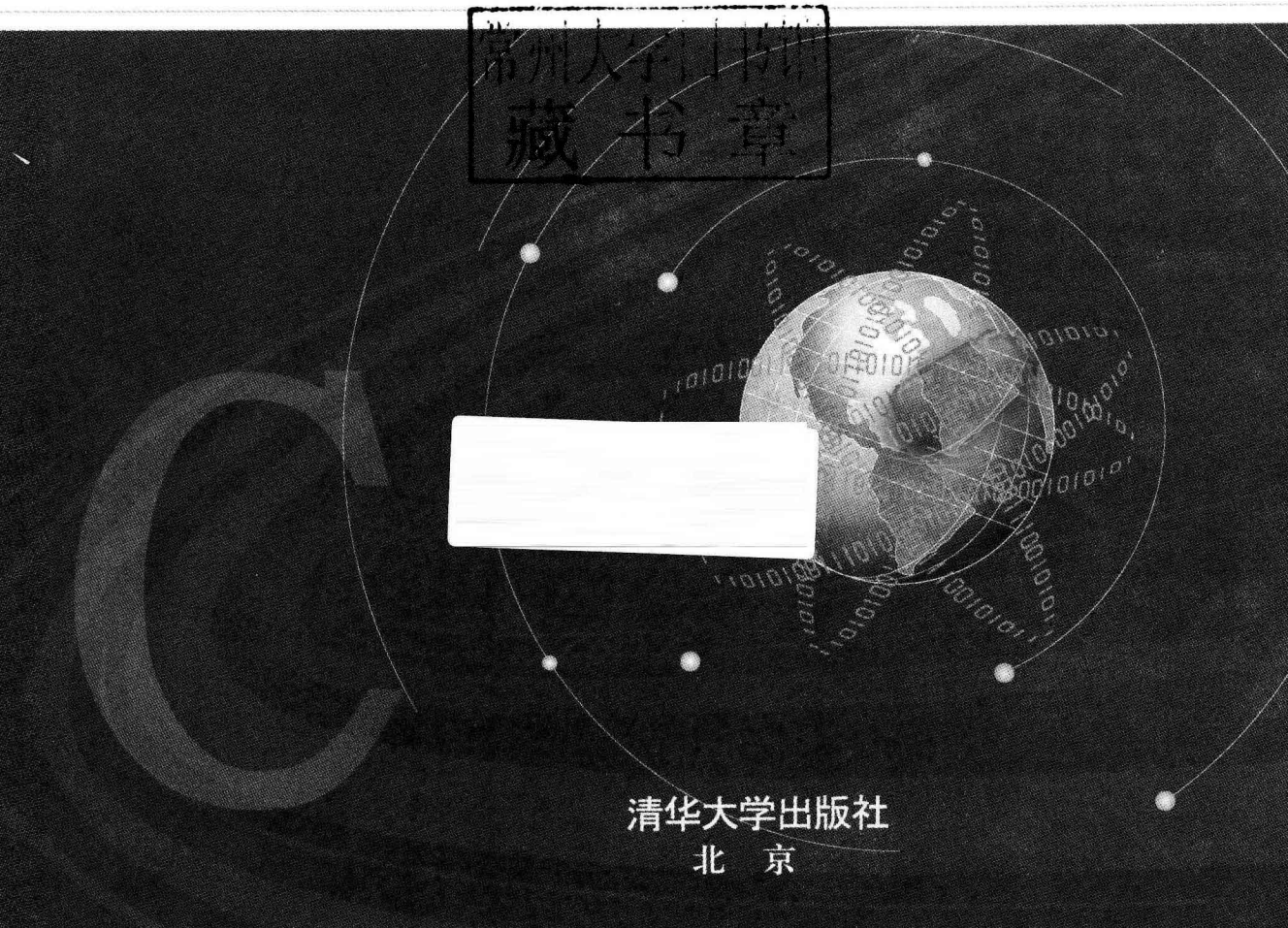
高潮　刘兴林　等编著



清华大学出版社

C语言程序设计

高潮　刘兴林　等编著



清华大学出版社
北　京

内 容 简 介

本书以C语言为编程工具，介绍了程序设计的基本概念和基本方法。全书不拘泥于语言细节，而是注重解题思路、算法实现和程序设计思想，在保证概念清晰、准确的前提下力求做到语言通俗易懂，引导学生真正进入程序设计的门槛，为后续专业课程的学习与计算机技术的实际应用打下良好的基础。

本书分为教程、实验指导和附录3个部分。教程部分由8章组成，包括概述、C语言基础、算法与控制结构、函数及编译预处理、数组与字符串、指针、构造数据类型和文件等内容。每章均配有一定量的思考题、选择题、填空题和编程题。在实验指导部分，设置了9个与教程相关章节配套的实验项目和一个综合实验项目。在附录中，提供了Visual C++6.0开发环境及程序调试、常用库函数等学习C语言程序设计的重要支撑内容，以及具有编程题自动评分功能的《C语言程序设计》测试与练习系统的介绍和使用说明。

本书配有PPT电子教案，例题、习题与实验的程序源代码，网络测试系统等教学资源，可免费向任课教师提供。

本书适合作为高等院校理工科各专业的程序设计课程教学用书，也可供从事计算机相关工作的技术人员、计算机爱好者及各类自学人员参考使用。

图书在版编目（CIP）数据

C语言程序设计/高潮，刘兴林等编著. —北京：清华大学出版社，2012.8

ISBN 978-7-302-29418-4

I. ①C…　II. ①高…　②刘…　III. ①C语言-程序设计　IV. ①TP312

中国版本图书馆CIP数据核字（2012）第161075号

责任编辑：朱英彪
封面设计：刘　超
版式设计：文森时代
责任校对：张兴旺
责任印制：杨　艳

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
印 装 者：保定市中画美凯印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：16.75　　**字　　数**：387千字
版　　次：2012年8月第1版　　**印　　次**：2012年8月第1次印刷
印　　数：1～5000
定　　价：29.80元

产品编号：048714-01

前　言

C 语言是一种被广泛应用的计算机高级语言，既可以用来开发系统软件，也可以用来开发应用软件，还能直接访问物理地址，具有位（bit）操作的功能，可以直接对硬件进行操作。这使得 C 语言既具有高级语言的所有优点，又具有汇编语言的许多功能。可以说，几乎没有不能用 C 语言实现的软件，几乎没有不支持 C 语言的系统。当前流行的、新生的计算机高级语言都借鉴了 C 语言的思想、语法和风格，各种需要使用计算机语言来描述有关内容的论文、专著、教材等，最常见的也是 C 语言或由 C 语言发展而来的 C++语言。即使不编程的人，通过学习 C 语言，也可以更好地了解计算机，了解计算思维。

本书以 C 语言为编程工具，介绍程序设计的基本概念和基本方法。全书分为教程、实验指导和附录 3 个部分。教程部分由 8 章组成，包括概述、C 语言基础、算法与控制结构、函数及编译预处理、数组与字符串、指针、构造数据类型和文件等内容。每章均配有一定量的思考题、选择题、填空题和编程题。在实验指导部分，设置了 9 个与教程相关章节配套的实验项目和一个综合实验项目。在附录中，提供了 Visual C++6.0 开发环境及程序调试、常用库函数等学习 C 语言程序设计的重要支撑内容，以及具有编程题自动评分功能的《C 语言程序设计》测试与练习系统的介绍和使用说明。

本书具有如下一些特点。

（1）教学学时的适应性强。本书内容中，*号标注的部分为选学内容，*号以外的部分是一个完整的体系。例如，基本例题已经直接与相关章节的内容有机地结合在一起，而提高性、综合性的例题则作为选学内容集中在每章最后一节的程序举例中。再如，对初学程序设计人员还不会迫切需要学习和应用的位运算、链表、文件的随机访问等内容也将作为选学内容。任课教师可根据教学学时的多少和专业的差异，选讲或不讲有关的选学内容，这不会影响对 C 语言程序设计的整体把握和系统学习。不包括*号标注的部分，教学学时只需 40 学时左右，包括全部*号标注的选学内容，大约需要 70 学时。

（2）系统性与灵活性相结合。在不影响整体系统性的前提下，将难点内容合理分解，将琐碎内容从教材主体中直接舍去。例如，学习 C 语言程序设计一般不需要掌握复杂的格式化输入/输出控制操作，更不应陷入到复杂的输入/输出格式控制及有关问题中，这样才有利于整体把握程序设计的思想和方法，因此在教材主体中对于输入/输出函数以“够用”为原则，在需要的地方仅做适当的介绍和说明，而将格式输入/输出函数的详细介绍放在附录中。另一方面，不能为了分解难度而牺牲系统完整性，不能把教材等同于体现教学过程的教案。例如，不宜把运算符与表达式分解到不同的章节中进行介绍，因为运算符与表达式的难点主要不在语法，而在于如何由数学思维向计算机思维转换。因此，我们要将运算符与表达式作为 C 语言程序设计学习的第一层次来介绍，这是一个由数学思维向计算机思维转换的重要阶段。其实，只要有初中数学知识的基础就可以理解大部分的 C 语言运算符与表达式。在介绍 C 语言运算符与表达式时，我们首先应该告诉学生，要在已有数学知识的

基础上，注意计算机语言与数学语言之间的联系和区别，注意计算机语言特殊的表达方式和具体要求，然后重点介绍 C 语言中与已有认知不同的内容，重点引导学生学会由数学语言表达式向计算机语言表达式进行转换，并在后续学习中自觉地理解和应用 C 语言的运算符和表达式去实现有关命题。

（3）注重 3 个层次逐级递进的 C 语言程序设计的学习过程，适当加强运算符与表达式的教学内容。

第一层次是学习如何由数学语言表达式向计算机语言表达式转换，如何将一个基本命题转换为 C 语言的表达式。这是一个容易被忽略但却非常重要的学习层次，必须加强该层次的学习。因为只有懂得了如何将一个基本命题通过 C 语言表达式写出来，才可能进一步运用 3 种基本控制结构实现基本的算法。第二层次是学习如何运用顺序、选择和循环 3 种基本控制结构实现结构化的算法，学习结构化程序设计的基本方法。第三层次是学习如何通过函数实现对功能（或过程）的封装，学习模块化程序设计的基本方法。

（4）注重解题思路、算法实现和程序设计思想，而不拘泥于语言，不纠缠于细节，尽量避免琐碎的描述和概念的堆砌，在保证概念准确的前提下力求做到语言通俗易懂。

（5）尽早引入函数概念，以建立程序模块思想，实现对功能（或过程）的封装，并注重建立函数的思考问题的方法，而不仅仅是介绍语法规则；注重在后续章节中以函数（功能模块）来组织程序代码，巩固算法的结构化思想及程序模块思想。

（6）实验指导与教程内容有机结合，并可借助于编程题自动测试系统，实现对学生编程学习情况的快速反应。学习程序设计，编程是硬道理，必须加强实验指导和编程训练，提高程序设计学习的实效性。

本书的第 1 章、第 2 章、第 5 章、上机实验指导和附录由高潮编写，第 3 章由吴明芬编写，第 4 章由李志仁和彭腊梅编写，第 6 章由彭腊梅编写，第 7 章由高宏宾编写，第 8 章由刘兴林编写。全书由高潮和刘兴林负责统稿。在本书的编写过程中得到了五邑大学计算机学院领导的大力支持，五邑大学计算机学院的有关老师也对本书的编写提出了许多宝贵的意见和建议，在此一并表示感谢。

本书配有 PPT 电子教案，例题、习题与实验的程序源代码，网络测试系统等教学资源，可免费向任课教师提供，有需要者请发电子邮件到 gordon911@126.com 索要。

由于编者水平有限，书中的错误及不妥之处在所难免，敬请读者给予批评与指正。

编　者

2012 年 6 月

目　录

第1章 概 述

本章首先通过一个现实的例子来说明有关程序的几个概念，再通过对软件危机的讨论来说明程序与软件的关系及软件开发过程，然后就面向过程的结构化程序设计思想、C 语言的发展及 C 语言程序的开发过程等进行简要介绍。

1.1 程序、算法、数据结构及程序设计语言

我们先看一个现实中的例子——“厨师做菜”。

厨师在开始做菜前，首先要知道这道菜需要哪些原料，各种原料如何搭配；还要知道如何在恰当的时机对相应的原料采取恰当的方法进行加工和处理。厨师所要知道的这些内容，其实就是一道菜的菜谱。厨师按照菜谱开始操作，最后做出一道可口的菜肴。

以上所谓的“原料如何搭配”说明的是操作的对象；“恰当的时机”、“恰当的方法”，也就是事先已经确定好的做菜的具体步骤和方法，即做菜的程序。这里的“程序”是人们普遍使用的一个朴素的概念，但也说明了这种思考问题的方法的核心是“过程”。

运用计算机来解决一个实际问题，与“厨师做菜”的道理是一样的：首先需要把问题处理的对象搞清楚，处理的具体步骤和方法要事先设计好。然后再利用计算机能够理解的语言写出具体的操作步骤，这就是计算机程序。计算机按照程序就可以自动执行程序指令，解决相应的问题，完成相应的任务。

从以上描述中，我们可以明确这样几个概念：程序、算法、数据结构和程序设计语言。

程序（Programs）就是要计算机完成某项工作的代名词，是对计算机完成某项工作所涉及的数据对象和动作规则的描述。**算法**（Algorithms）体现在动作规则的描述中，它描述了解决一个问题所采取的方法和步骤。**数据结构**（Data Structure）体现于对对象（或数据）的描述，它描述了问题所涉及的对象以及对象之间的联系和组织结构。数值计算问题可以用数学方程来描述，但更多的非数值计算问题无法用数学方程加以描述，而是通过表、树和图之类的数据结构来建立数学模型。

1976 年，著名计算机科学家沃思（Nikiklaus Wirth）出版了一本名为《Algorithms+Data Structure=Programs》的著作，明确提出算法和数据结构是程序的两个要素，即：

程序 = 算法 + 数据结构

也就是说，程序设计主要包括两方面的内容：行为特性的设计和结构特性的设计。行为特性的设计是指完整地描述问题求解的全过程，并精确地定义每个解题步骤，这一过程即算法的设计；而结构特性的设计是指在问题求解的过程中，计算机所处理的数据及数据之间联系的表示方法。程序设计的关键是构造程序的数据结构，并描述问题所需要的、施加在这些数据结构上的算法。

程序最终需要使用计算机能够理解的语言具体地写出来，这就是计算机程序设计语言。所谓“语言”就是一套具有语法、词法规则的系统。语言是思维的工具，思维是通过语言来表述的。**计算机程序设计语言**（Computer Programming Language）是计算机可以识别的语言，是程序实现的工具。任何一门计算机程序设计语言都需要通过数据类型、运算符与表达式以及控制语句等来定义和实现程序中的数据结构和算法。

1.2 程序与软件及软件开发过程

在计算机应用的初期，人们狭义地认为软件就是程序，软件设计就是程序设计，软件是程序员个人劳动的成果。然而在 20 世纪 60 年代以后，随着计算机系统性能的提高和应用范围越来越广泛，计算机软件系统的开发也变得越来越复杂，在开发大型软件的过程中出现了所谓的“软件危机”，其主要表现是：开发进度被推迟，成本超出预算，软件产品的可靠性降低等。人们认识到开发软件要比想象的复杂得多。

为什么会出现软件危机呢？其主要原因有 3 个。

（1）软件开发者与用户之间对问题的理解不一致。一方面开发者不熟悉用户问题的领域，或没有理解用户的需求，导致设计的软件产品与用户要求不一致；另一方面，用户难以提出准确、完整、无二义性的软件需求描述。

（2）软件开发过程更多地体现为设计人员个人的思考过程，这种思考过程不可能完整地表现在书面上。因此无法对思考过程进行科学规范及质量管理，软件开发进度也无法控制。

（3）人的智力在面对越来越复杂的问题时，处理问题的效率会越来越低。因此，在没有找到控制问题复杂性的有效方法时，开发软件所需的时间和费用将随问题复杂性的增加而急剧增加。

经历了“软件危机”之后，人们对软件的概念和软件开发过程有了新的认识，即软件不仅仅是程序，软件开发也不仅仅是程序设计。计算机软件是指计算机程序和与之相关的文档资料的总和，文档资料包括编制程序所使用的技术资料和使用该程序的说明性资料（如使用说明书等），即开发、使用和维护程序所需的一切资料，如图 1-1 所示。计算机软件开发则是一个包括程序设计在内的可以进行管理、控制和质量评价的规范的工程。“软件工程”的概念由此而生。

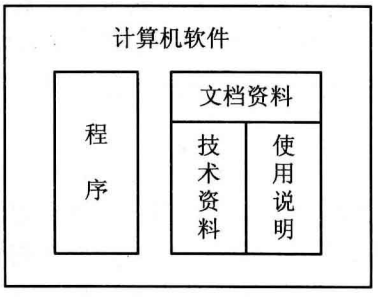


图 1-1　计算机软件概念示意图

简单地说，一个大型软件系统的开发，必须经历这样一个过程：

需求分析→系统设计→程序编码以及编辑、编译和连接→系统测试→运行维护

在软件开发过程的每一个阶段，都要有明确的任务和要求，必须产生一定规格的文档资料。

1.3　面向过程的结构化程序设计

软件开发过程中最关键的是需求分析和系统设计，然后再通过具体程序编码去实现。而系统设计中的一个关键内容是选择合适的软件开发方法。在如何克服“软件危机”的讨论和研究中，诞生了“结构化程序设计”的概念，它是一个面向过程的软件开发方法，它的产生和发展形成了现代软件工程的基础。

结构化程序设计的基本思想是：自顶向下、逐步求精，将一个复杂的问题分解为若干个易于处理的子问题，从而将整个程序划分成若干个功能相对独立的子模块或过程。子模块又可继续划分，直至最简。每个模块都只有一个入口和一个出口；整个程序则全部可以用顺序、选择和循环这 3 种基本结构及其组合来实现。这样的结构化程序设计，以解决问题的过程作为程序的基础和重点，是一种面向过程的程序设计方法。

在面向过程的结构化程序中，程序是过程模块化的，模块是分层次的，层与层之间是一个从上往下的调用关系，如图 1-2 所示。面向过程程序的基本构成单位是过程（或者函数）。

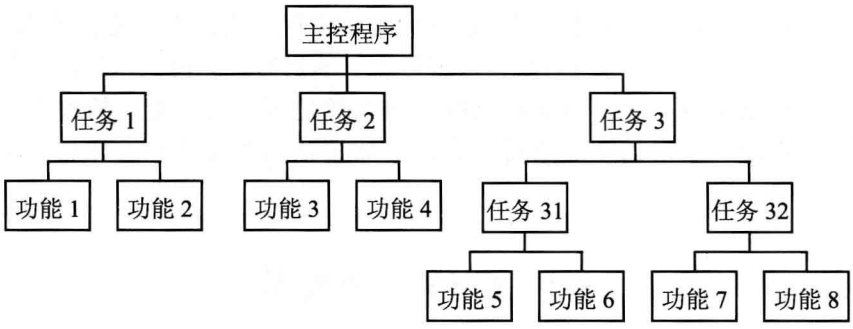


图 1-2　面向过程程序的程序结构

下面来看看一个餐馆的整个运营过程——“餐馆运营系统”：顾客到餐馆吃饭，服务员负责接待顾客；顾客就座后，点好要吃的饭菜，服务员就去告诉厨师要做哪些菜；厨师做好以后，服务员端上饭菜，顾客就可以品尝这美味佳肴了；最后顾客吃完饭付清账款。

我们可以把这个系统按功能抽象分解为点菜的过程、做菜的过程、品尝菜肴的过程、收付账款的过程等，如图 1-3 所示。这样来组织系统的方式以过程为核心，是面向过程的。

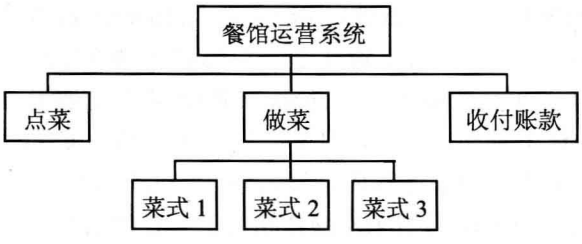


图 1-3　餐馆运营系统（面向过程）示意图

面向过程的结构化程序设计方法有许多优点。

（1）各模块可以分别编写，使得程序更易于阅读、理解、测试和修改。这样不管编多大的程序，不管有多少人参加编写，都可以把他们的模块有效地连接起来。

（2）方便增加新的功能模块。

（3）功能独立的模块可以组成子程序库，有利于实现软件的复用。

但另一方面，由于结构化程序设计方法强调的是模块的功能，是面向过程的，数据与处理这些数据的过程是分离的。这样的系统设计方法也会带来如下一些问题。

（1）对不同格式的数据作相同的处理，或是对相同的数据作不同的处理，都需要编写不同的程序模块来实现，使得程序的可复用性大打折扣。

（2）由于过程和数据相分离，数据可能被多个模块所使用和修改。这样很难保证数据的安全性和一致性。

（3）当数据处理的方法或是数据类型发生改变时，会导致整个系统的重新设计和编码。

由于上述问题，面向过程的结构化程序设计方法难以适应大型软件的设计开发。为此，能够克服上述问题的面向对象的软件开发方法，在20世纪80年代以后得到了重视并快速发展，已成为当前软件开发方法的主流。但是面向对象的程序设计方法与面向过程的结构化程序设计方法不是对立的，面向对象的程序设计在技术方法上是对传统软件开发方法的继承和发展，它汲取了面向过程的结构化程序设计中最为精华的部分，即以顺序、选择和循环3种结构实现对功能（或过程）的封装。面向过程与面向对象的程序设计都要以算法设计为基础，算法和数据结构仍然是各种系统建立的根本。在学习程序设计的基础阶段，学习的重点还是在算法与数据结构以及过程设计与封装的基本技术方法上。

1.4 C语言的发展

C语言是一种广泛流行的计算机高级语言，既可以用来开发系统软件，也可以用来开发应用软件。

在C语言诞生以前，系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差，但一般的高级语言又难以实现对计算机硬件的直接操作（这正是汇编语言的优势），于是人们盼望有一种兼有汇编语言和高级语言特性的新语言。

1970年，美国AT&T的贝尔（Bell）实验室的K.Thompson以BCPL（Basic Combined Programming Language）语言为基础，设计了一种类似于BCPL的语言，取其第一字母B，称为B语言。1972年，贝尔实验室的Dennis M.Ritchie为克服B语言的诸多不足，在B语言的基础上重新设计了一种语言，取其第二字母C，称为C语言。BCPL和B语言最初都是为编写操作系统软件和编译器而开发的语言。贝尔实验室在B语言的基础上开发出的C语言，最初也是用来编写UNIX操作系统的，C语言作为UNIX操作系统的开发语言为人们所认识。但由于C语言严格的设计，与具体硬件无关的特性以及其他优点，

使C语言的应用很快就超出了贝尔实验室的范围。20 世纪 70 年代末，C 语言开始移植到非 UNIX 环境中，并逐步脱离 UNIX 系统成为一种独立的程序设计语言，迅速地在全球传播。

C 语言之所以备受青睐，是和它具有的许多优点分不开的。

（1）C 语言简洁紧凑，只有 32 个关键字和 9 种控制语句，而且书写形式自由，比较容易入门。

（2）丰富的数据类型。C 语言不仅有基本类型，而且还有组合类型、指针类型等，能用来实现各种复杂的数据结构。

（3）丰富的运算符。C 语言的运算符相当多，许多操作都可以用运算符表示。由运算符和运算对象可以组成表达式，因而 C 语言中表达式的类型也是相当丰富的。

（4）具有结构化的控制语句。顺序、选择和循环这 3 种类型的结构化控制，在 C 语言中都有相应的语句加以体现，因此可以用 C 语言进行结构化程序设计。C 语言还以函数作为程序的基本单位，便于实现程序的模块化和过程（功能）的封装。

（5）C 语言作为一种高级语言，能直接访问物理地址，具有位（bit）操作功能，可以直接对硬件进行操作。这使得 C 语言既具有高级语言的所有优点，又具有汇编语言的许多功能。

（6）用 C 语言编写的程序编译后所生成的目标代码质量高，程序的执行效率高。

因为上述优点，C 语言既可以用来开发系统软件，也可以用来开发应用软件。但另一方面，由于 C 语言的语法限制不太严格，程序设计自由度大，所以对程序员要求也较高。而且 C 语言是一种面向过程的程序设计语言，面向过程的程序设计是以过程处理为核心，这种设计方法有其局限性。

自 20 世纪 80 年代初开始，随着面向对象程序设计思想的日益普及，很多支持面向对象程序设计方法的语言也相继出现了，C++就是其中之一。C++是 Bjarne Stroustrup 于 1980 年在 AT&T 的贝尔实验室开发的一种语言，它是 C 语言的超集和扩展，是在 C 语言的基础上扩充了面向对象的语言成分而形成的。最初这种扩展后的语言称为带类（class）的 C 语言，1983 年才被正式称为 C++语言，之后又经过不断的改进，发展成为今天的 C++语言。

C++语言与 C 语言完全兼容，很多用 C 语言编写的库函数和应用程序都可以为 C++语言所用。但正是由于与 C 语言兼容，使得 C++语言不是纯粹的面向对象的语言，它是一种既支持面向对象也支持面向过程的程序设计语言。

由于 C 语言的广泛应用，C++语言又对 C 语言具有的向下兼容性，以及 C++语言对面向对象技术的支持，C++语言推出后，很快就获得商业上的成功。各大公司纷纷推出自己的 C++语言编译器和开发工具，其中微软公司的 Visual C++是目前最流行的 C++语言开发平台之一。

微软公司的Visual C++系统同时具有C++语言编译器和C语言编译器，既可以开发C++语言程序，也可以开发 C 语言程序。本书将以 Visual C++6.0 为平台来介绍 C 语言程序设计的基本原理和方法。

1.5 C语言程序的开发过程

1.5.1 几个基本术语

（1）源程序

用计算机程序设计语言编写的程序称为**源程序**（source program）。但计算机还不能直接识别源程序代码，这些代码有待被“翻译”成计算机能够识别的二进制的程序代码。C语言的源程序代码文件，其文件名一般以.c作为后缀。

（2）目标程序

为了使计算机能执行高级语言编写的源程序，必须先用一种翻译程序，把源程序翻译成二进制形式的目标程序（object program）。二进制形式的**目标程序**又称为机器语言代码程序，是计算机能直接识别的程序。

（3）翻译程序

翻译程序是指一个把源程序翻译成等价的目标程序的程序。

翻译程序有3种不同类型：汇编程序、编译程序和解释程序。

- 汇编程序

将汇编语言写成的源程序，翻译成二进制代码的目标程序，这种翻译过程称为**汇编**，实现这种翻译任务的程序称为**汇编程序**。

- 编译程序

对于用高级程序设计语言编写的源程序，若是以源程序文件为单位，整体翻译成二进制目标程序文件，这种翻译过程称为**编译**，实现这种翻译任务的程序称为**编译程序**。编译程序又常常被称为**编译器**（complier）。

C语言是一种编译型的语言。C语言的源程序文件经由编译器加工生成的目标程序文件，其文件名一般以.obj或.o作为后缀（object的缩写）。

- 解释程序

解释程序的任务，同样也是将高级语言源程序翻译成二进制目标代码，但与编译程序不同的是，解释程序是边翻译边执行，即输入一句源程序代码、翻译一句、执行一句，直至将整个源程序翻译并执行完毕。

显而易见，与编译型语言相比，用解释型语言编写的程序，其执行效率和运行速度要低一些。

（4）连接程序

用编译型语言编写的源程序经由编译器加工生成的目标程序文件，还不是计算机可以直接执行的程序，还要用系统提供的**连接程序**将一个程序的所有目标文件和系统的库文件以及系统提供的其他信息连接起来，最终形成一个计算机可直接执行的二进制程序文件。

连接程序又常常被称为**连接器**（linker）。连接生成的可执行文件，其文件名的后缀是.exe。

1.5.2　开发 C 语言程序的基本过程

C 语言是一种编译型的语言，其程序的开发过程与其他编译型语言一样，通常要经过源程序的编辑、编译、连接和运行调试这几个步骤，如图 1-4 所示。图中实线表示操作流程，虚线表示文件的输入/输出。

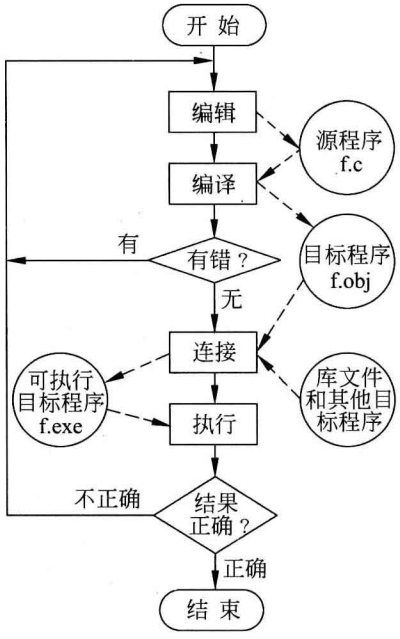


图 1-4　开发 C 语言程序的基本过程

现在市场上有很多 C 语言的编译器和开发工具。本书将以目前最流行的微软公司的 Visual C++ 6.0 语言为平台来介绍和学习 C 语言程序设计。Microsoft 公司的 Visual C++ 系统提供了一个基于 Windows 平台的可视化的集成开发环境（IDE，Integrated Development Environment），它集程序的编辑、编译、连接和运行调试等功能于一体，而且提供了更加强大的系统集成能力。关于 Visual C++6.0 语言开发环境的进一步认识和使用，请参见“实验一　初识 C 语言程序开发环境”及“附录 F　Visual C++ 6.0 开发环境及程序调试”。

习　题　一

思考题

1. 什么是程序？什么是软件？程序与软件的关系如何？

2. C 语言有哪些特点？C 语言与 C++语言是怎样一种关系？
3. 本章的“C 语言程序的开发过程”与“软件开发过程”是什么关系？开发一个 C 语言程序的一般步骤是什么？
4. 在一个 C 语言程序开发过程的关键步骤中，相应产生了哪几种文件？文件的扩展名是什么？