



高等院校计算机科学与技术规划教材



C 语言

程序设计实践训练教程

朱承学 余肖勇 主编
王新辉 李 崧



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校计算机科学与技术规划教材

C 语言程序设计实践训练教程

朱承学 余肖勇 王新辉 李崧 主编

中国水利水电出版社

内 容 提 要

计算机实践训练课程是程序设计课程中必不可少的一个环节。目的是训练读者综合运用该课程的知识与技巧的能力，达到理论联系实际、学以致用。

本书是与《C 语言程序设计教程》和《C 语言程序设计实验指导与习题解析》配套使用的教学参考书。内容包括：程序设计的基本理论与方法、C 语言实用程序设计技术、实例分析、参选课题。

本书中的示例都已上机通过。示例选材恰当，分析深入浅出、循序渐进，具有启发性和实用性，与理论教学紧密配合。

本书是 C 语言实践训练的必备参考书，可以作为高等院校计算机专业或非计算机专业的计算机程序设计实践教学用书，也可作为计算机应用科技人员的参考书或培训教材。

本书实例源代码可以从中国水利水电出版社网站（www.waterpub.com.cn）上下载。

图书在版编目（CIP）数据

C 语言程序设计实践训练教程 / 朱承学等主编. —北京：中国水利水电出版社，2004

（21 世纪高等院校计算机科学与技术规划教材）

ISBN 7-5084-2131-0

I. C… II. 朱… 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（2004）第 076486 号

书 名	C 语言程序设计实践训练教程
主 编	朱承学 余肖勇 王新辉 李崧
出版 发行	中国水利水电出版社（北京市三里河路 6 号 100044） 网址： www.waterpub.com.cn E-mail： mchannel@263.net （万水） sales@waterpub.com.cn
经 售	电话：（010）63202266（总机）、68331835（营销中心）、82562819（万水） 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京北医印刷厂
规 格	787mm×1092mm 16 开本 12.5 印张 284 千字
版 次	2004 年 9 月第 1 版 2004 年 9 月第 1 次印刷
印 数	0001—5000 册
定 价	18.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

版权所有·侵权必究

21 世纪高等院校计算机科学与技术规划教材

编委会名单

主任 陈火旺 中国工程院院士 国防科技大学教授

委员 周立柱 清华大学计算机系主任

杨放春 北京邮电大学计算机科学与技术学院院长

杨学军 国防科技大学计算机学院院长

徐晓飞 哈尔滨工业大学计算机科学与技术学院院长

李仁发 湖南大学计算机与通信学院院长

卢正鼎 华中科技大学计算机学院院长

李志蜀 四川大学计算机学院院长

戴居丰 天津大学信息学院、软件学院院长

蒋昌俊 同济大学计算机科学与工程系主任

何炎祥 武汉大学计算机学院院长

周兴社 西北工业大学计算机系主任

陈志刚 中南大学信息科学与工程学院副院长

姜云飞 中山大学软件学院院长

周昌乐 厦门大学软件学院院长

齐勇 西安交通大学计算机科学与技术系主任

赵书城 兰州大学计算机学院院长

孟祥旭 山东大学计算机学院院长

前 言

计算机是一门实践性较强的学科，实践环节的教学和训练非常重要，缺少实践环节的计算机教学往往是空洞而枯燥的，并会导致教学目标难以实现，造成教学资源的浪费。程序设计课程尤为突出。

计算机实践训练课程是继计算机程序设计语言课程之后的又一门技术基础必修课程，一般以课程设计的形式开设。通过计算机实践训练课程的学习，能巩固所学程序设计语言的基本知识，培养学生的科学实践能力，培养学生利用计算机分析和解决实际问题的能力，训练学生将计算机技术应用到人们的生产生活中，提高社会生产率，提高人们的生活品质。但是，由于计算机实践课程目前还处于一种摸索阶段，到目前为止尚未形成规范的教学、实践和考核模式，各种教学和学习资源没有得到充分的利用，有很多的实践训练课程没有达到应有的效果。为了进一步完善计算机基础教学，规范 C 语言程序设计实践训练课程的教学与实践，作者综合多年从事计算机实践课程教学和指导的经验，组织我校多年来一直参与计算机实践课程指导的教师编写了这本实践训练教程。

本书共分为 4 章，第 1 章阐述了程序设计的一些基本概念，让读者对与程序有关的概念有一个基本的了解。第 2 章介绍了利用 C 语言开发综合应用程序时常用的一些技术和方法，为读者开发实用的 C 语言程序提供技术上的支持。第 3 章详细描述三个常见问题的分析与解题过程，并给出参考程序，便于读者了解并掌握利用 C 语言开发应用程序的方法和技巧。第 4 章列出作者多年来布置给学生的一些实践课题及提示，供指导教师和学生参考。书中所给程序均在 Turbo C 2.0 及 Turbo C 3.0++ 环境下运行通过。

本书由朱承学、余肖勇、王新辉、李崧组织编写，参加本书编写的还有安永锋、李劲强、李梦晖、裘嵘、徐彤、刘群芳、成海英、孙岱等。另外，我的学生谢秀峰、刘树波为书中部分程序的调试做了大量的工作，在此表示感谢。

在本书的编写过程中，得到了蒋外文教授和刘卫国教授的热情支持与指导，在此表示衷心感谢。

由于时间仓促及作者水平有限，书中疏漏和不当之处在所难免，恳请广大读者批评指正。

编 者

2004 年 5 月

目 录

前言

第 1 章 程序设计课程实践训练概述	1
1.1 学习程序设计的目的与任务	1
1.2 实践训练的必要性	2
1.3 软件工程概述	2
1.3.1 软件危机与软件工程	2
1.3.2 软件工程的基本原理	3
1.3.3 软件的生命周期	3
1.4 程序设计的基本方法和步骤	4
1.4.1 利用计算机处理问题的步骤	4
1.4.2 结构化程序设计技术	5
第 2 章 C 语言实用程序设计技术	8
2.1 图形处理基础	8
2.1.1 图形程序运行的条件	8
2.1.2 图形显示器与适配器	9
2.1.3 图形显示的坐标和像素	10
2.1.4 基本图形处理函数及图形处理技术	11
2.2 菜单设计	23
2.2.1 菜单	23
2.2.2 菜单设计基础	24
2.2.3 菜单设计实例	26
2.3 屏幕图形的存取	44
2.3.1 屏幕图形与 VRAM 地址的关系	44
2.3.2 VRAM 的位面结构和读写操作	45
2.3.3 存取屏幕图形时地址指针的设置	50
2.3.4 将屏幕图形存入文件	51
2.3.5 将图形文件显示到屏幕	55
2.4 动画技术	58
2.4.1 利用画擦法实现简单动画	58
2.4.2 利用存储再重放方法	60
2.4.3 利用显示页和编辑页交替变化方法设计动画	61
2.4.4 直接对图像动态存储进行操作方法	64

2.5	I/O 接口的输入输出	69
2.5.1	I/O 接口的寻址方式	69
2.5.2	I/O 接口常用函数	70
2.5.3	喇叭发声程序	71
2.5.4	异步通信	74
2.6	中断服务程序的设计	76
2.6.1	PC 机中断概述	76
2.6.2	中断程序的编写方法	78
2.6.3	有声动画程序	79
2.6.4	采用中断方式的数据采集程序	82
2.7	驻留程序的设计	84
2.7.1	驻留程序设计概述	84
2.7.2	TSR 程序设计	84
2.7.3	键盘处理	86
2.7.4	使用鼠标	88
2.8	汉字显示技术	96
2.8.1	在西文 DOS 下 C 程序显示汉字技术概述	96
2.8.2	用直接写显示存储器 VRAM 的方法显示汉字	97
2.8.3	利用 C 语言函数显示汉字	103
2.8.4	汉字的放大技术	104
2.8.5	小型专用汉字库的建立方法	106
第 3 章	典型课题分析与解答	110
3.1	简易学生成绩管理系统的设计与实现	110
3.1.1	设计要求	110
3.1.2	预备知识	110
3.1.3	问题的分析与设计	111
3.1.4	编码与实现	116
3.2	电子时钟系统的设计与实现	136
3.2.1	设计要求	136
3.2.2	预备知识	137
3.2.3	问题的分析与设计	137
3.2.4	编码与实现	142
3.3	简易动画——单摆运动	147
3.3.1	设计要求	147
3.3.2	预备知识	147
3.3.3	问题的分析与设计	148
3.3.4	编码与实现	153

第4章 课题精选与提示	162
附录1 Turbo C库函数一览表	171
附录2 图形模式表	189
附录3 鼠标器功能及功能编号	190
参考文献	191

第 1 章 程序设计课程实践训练概述

知识点:

- 程序设计的方法与步骤
- 软件开发基础

本章导读:

本章主要介绍程序设计的方法与步骤及软件工程的一些基本思想,让读者对程序设计及实践训练课程有一个正确的认识。

1.1 学习程序设计的目的与任务

著名学者 Christopher Strachey 曾经说过“如果一个人不能理解程序设计语言,他就不能真正地理解计算机。理解程序设计语言并不是仅仅意味着会使用它们,许多人能使用它们,但是并不真正理解。”这表明了程序设计课程的重要性以及真正掌握该课程是有一定难度的。

目前,受实际情况的限制,很多程序设计课程的教学仍然停留在让学生学习一门语言的层次上,更有一些教学是为了完成某一个教学过程。据笔者调查,不少学生认为该课程没有任何价值。一方面,学习过程中,不明确为什么而学,所学知识有什么应用,只是觉得应该可以用来编写程序,但是什么是程序,怎样编写程序,心中非常茫然;另一方面,正因为学习过程是一头雾水,课程结束以后,头脑中没有丝毫印象。因此,不少学生把该课程理解为挣学分的需要或者是取得某种证书的需要。

其实,程序设计课程教学的根本目的不是为了使学生熟悉一门语言,掌握一种编程工具,而应该是让学生掌握程序设计的基本思想和方法,培养学生综合利用某种语言进行程序设计的能力,培养学生利用系统提供的功能进行创新设计的能力,培养学生充分利用计算机的功能和特点分析实际问题和解决实际问题的能力,最终将计算机技术应用到我们的生产生活中去,提高社会生产率和我们的生活品质。

事实上,学生经过一个学期的程序设计课程的学习,了解并掌握了一门程序设计语言的语法规则,熟悉了简单程序的设计方法和编写,但对程序设计及该门程序设计语言的应用的理解还是肤浅的或模糊的。计算机课程设计正是应这种情况而设置的。

计算机课程设计是计算机基础教学中的一个重要环节,是继计算机文化基础和计算机程序设计课程之后的又一门技术基础必修课。通过模拟的或实际课题的教学与训练,为学生提供一个综合运用所学知识解决实际问题的机会,增强学生的实践动手能力和工程实践

能力，并培养和锻炼学生的自学、创新能力。

1.2 实践训练的必要性

由于我国计算机文化素质教育普及相对较晚，绝大多数大专院校的学生对计算机的了解还是很肤浅的。对于大学一年级或二年级的学生，尽管经历了计算机文化基础和程序设计语言两门课程的学习，但是在实际生活中接触或运用得非常少，对计算机的理解只能是一知半解。人们学习和了解计算机，是因为它已经是人们学习、生活、工作中不可缺少的工具，因此学习它的根本的目的还是掌握它，并与生活、生产实际相结合，提高我们的生活水平和质量，提高我们的社会生产效率。要达到这个目的，课堂教学是远远不够的，必须将课堂教学与实践相结合，投身于一些实际系统的开发与研制，亲身了解实际问题的提出、分析、设计、实现的每一个环节。使学生从实践中领会程序设计课程学习的意义，并从实践中体会到学习的乐趣。最终认识到将计算机技术应用到各专业领域中的重要性和必要性，增强学习计算机技术的兴趣。

总而言之，实践是计算机学习的特点，也是计算机学习的根本目的。缺少实践课程的计算机学习是不完善的，甚至是毫无意义和毫无价值的。

1.3 软件工程概述

1.3.1 软件危机与软件工程

1968年，为了解决软件开发中出现的严重问题，在北大西洋公约组织的学术会议上第一次提出了“软件危机”和“软件工程”，并对软件工程化的技术进行了讨论。

软件危机是指在计算机软件开发和维护过程中所遇到的一系列严重问题。人们发现，为软件开发投入了大量人力和物力，所取得的系统的质量却差强人意。其主要表现有：

(1) 对软件开发成本和进度的估计不准确。实际成本往往远比估计成本高，开发周期往往一拖再拖。而为了赶进度和节约成本所采取的一些权宜之计又损害了软件的质量，从而严重影响软件开发组织的信誉并引起用户的不满。

(2) 用户对开发出的软件的满意率低。由于软件开发人员和用户之间缺乏交流，软件开发人员对用户的真正需求了解不够详细，从而“闭门造车”，最终导致软件与用户需求不吻合。

(3) 软件的质量不可靠，可维护性差。

(4) 软件缺少必要的文档资料，给软件的修改和维护带来了许多严重的困难和问题。

(5) 软件开发生产率的提高远远跟不上计算机应用的普及和发展的需要。软件开发正处于“供不应求”时期。

产生软件危机有两方面的原因：一方面与软件本身的特点有关，另一方面与软件开发和维护的方法不正确有关。软件的一个基本特点是规模庞大，一个软件往往有几百万行、

几千万行，甚至上亿行的目标代码，其复杂程度是可想而知的。这样复杂的系统，总是需要成百上千的技术人员长时间协同工作。这就涉及许多技术问题，如分析方法、设计方法、形式描述方法等，更重要的是怎样去管理这些技术人员协同工作。同时，一个程序在计算机上试运行之前，其开发进度、开发质量都难以评价。而试运行本身不可能以所有可能的数据对软件进行测试，也就不可避免地在正式投入运行后还会发现更多的错误。另外，用户对软件需求的不确定性或盲目性使开发者不能完整地把握用户的真正需要，从而导致开发结果不能满足实际需要或开发不成功。

要解决软件危机，不仅要有技术措施，还要有必要的组织管理措施。软件工程就是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门新兴学科。它采用工程的概念、原理、技术和方法来开发和维护软件，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，指导软件的开发与维护。

1.3.2 软件工程的基本原理

软件工程作为一门指导计算机软件开发和维护的工程学科，综合许多学者的研究和众开发公司的开发经验，归纳出了七条基本原理。

(1) 用分阶段的生命周期计划严格管理。根据这个原理，将软件开发分成若干个阶段，并制定每个阶段的可行性计划，然后严格按照计划对软件的开发与维护工作进行管理。

(2) 坚持进行阶段评审。在每一个阶段都要进行严格的评审，检查是否达到预定的目标，以便及时发现软件开发中出现的错误。

(3) 实现严格的产品控制。在软件开发过程中不应随意地根据用户的需要而修改方案和方法。对于一些必须的修改应进行严格的控制，以保持软件各个配置成分的一致性。

(4) 采用现代程序技术。采用各种结构分析(SA)和结构设计(SD)技术，提高软件开发的效率和软件维护的效率。

(5) 结果便于审查。软件产品本身是看不见摸不着的逻辑产品。为了提高软件开发过程的可见性，方便管理，根据软件开发项目的总目标及任务期限规定开发组织的责任和标准，便于能够清楚地检查所得到的结果。

(6) 开发人员少而精。开发人员的素质和数量是影响软件产品质量和开发效率的重要因素。这就要求软件开发人员的素质要好，且人数不应过多。

(7) 承认不断改进软件工程实践的必要性。新知识、新技术不断涌现，我们要积极主动地采纳新的软件技术，要不断地总结经验与教训，并不断地开发和研究新的软件技术。

1.3.3 软件的生命周期

一个软件从定义、开发、使用和维护，直到最终被废弃，要经历一个漫长的时期，我们通常把这个漫长的时期称为生命周期。一般说来，又将软件生命周期划分为几个阶段。根据软件规模、种类、开发方式、开发环境、开发方法等的不同，软件生命周期的划分方法有多种。一般来说，可以将软件生命周期划分为软件定义、软件开发和软件维护三大阶段，每一个阶段又可以划分若干个小阶段。每个阶段都有基本任务和结束标准。

软件定义阶段又可以称为系统分析阶段，可以分为三个子阶段：问题定义、可行性研究和需求分析。其任务是确定软件开发工程的总目标、确定软件开发的可行性、确定实现系统应该采用的策略、确定系统实现的功能、估算系统开发所需的资源及成本。

软件开发阶段又可以称为系统设计与实现阶段，完成对前一阶段定义的软件的具体设计与实现，通常分为四个子阶段：总体设计、详细设计、编码和单元测试、综合测试。其任务是：确定解决方案及相应的成本和效益、对确定的方案给出详细的规格说明、选取适当的程序设计语言编写程序并测试、将经过单元测试的模块按某种策略装配起来进行测试等。

软件维护阶段的主要任务是通过各种必要的维护活动使软件持久地满足用户的需求。维护通常分为四类：改正性维护，即诊断和改正在使用过程中发现的软件错误；适应性维护，即修改软件以适应环境的变化；完善性维护，即根据用户的需要改进或扩充软件使它更完善；预防性维护，即修改软件为将来的维护活动预先做准备。

1.4 程序设计的基本方法和步骤

要编写出一个好的程序，需要好的方法并实施正确的步骤。软件工程是进行开发工作的指南。当然，对于编写简单的程序，也许不需要考虑太多。一个小程序，从确定、编写、使用、修改是由同一个人来完成的，因此只要他一个人心里明白程序的思路就够了。但是，现在人们所面临的、需要解决的问题不是一个简单的算法，而是许许多多简单问题的有机组合。怎样把一个复杂的问题分解为许许多多多个小问题，然后再进行有效的组合，需要一定的方法和策略。而且，一个复杂问题的解决，往往也不是一个人可以完成的。注意，本节所讨论的问题还是属于简单问题，可以参照软件工程的方法，但不必完全照本宣科，毕竟我们的课程设计是为一个学生或一小组学生而设计的。而且，绝大多数的设计任务还只能是模拟的，并不要求其实用性，根本目的还是训练学生的能力。

1.4.1 利用计算机处理问题的步骤

无论使用计算机解决一个简单问题还是一个复杂问题，都必须事先设计好处理方案，然后编制好相应的程序，最后计算机按照相应的程序执行有关操作，得到所需结果。简单地说，求解一个实际问题，一般要经过以下几个步骤：

(1) 分析问题。确定已知数据、输出结果、所需的硬件和软件环境、提出求解问题的数学模型与计算方法。

(2) 算法分析。确定求解问题的数据对象和求解步骤，即给出算法。

(3) 程序设计。使用合适的程序设计语言把已确定的数据和算法描述出来，即编写程序，也称为编码。

(4) 程序调试。将编制的程序输入计算机系统中，进行编译和试运行，修正其中可能存在的语法和逻辑错误。

(5) 运行与维护。经调试后的程序基本正确，可以投入使用。但程序中的有些错误

并不一定能在调试中被发现,用户在使用过程中会发现一些隐藏的错误,需要维护人员进行修改。同时,技术的更新也需要维护人员对该程序进行改进。

(6) 整理输出结果,写出有关文档资料。

1.4.2 结构化程序设计技术

“结构化”的概念最早用来描述结构化程序设计方法,后来将它延伸到软件开发过程。结构化在程序设计中可以分为三个层次来理解:

(1) 具有结构化的算法设计,便于通过高级语言表达式的反复执行来完成某个算法。

(2) 结构化的数据类型,如数组、结构体、共用体、文件就是复杂的构造数据类型,可以满足各种复杂数据的描述。

(3) 结构化的语句编码,实现程序逻辑控制。

结构化程序设计是一种程序设计技术,它采用自顶向下逐步求精的设计方法和单入口单出口的控制结构。工作实践证明,采用结构化程序设计技术,不仅可以提高软件的质量,而且也大大提高了软件生产率。

1. C 程序设计的四种基本结构

C 程序由四种基本结构组成:顺序结构、选择结构、循环结构和函数。

(1) 顺序结构。这是一种最基本、最简单的按照语句顺序先后执行的结构。它由若干块组成,各块之间按照排列的先后顺序依次执行。一个块,指的是不含无条件转移(goto)语句,由赋值语句、函数调用语句、空语句等组成的程序段。

(2) 选择结构。选择结构是根据给定的条件决定选择执行语句。如 if 语句、if...else 语句和 switch 语句,尽可能不使用 goto 语句。

(3) 循环结构。循环语句实现某个程序段的重复执行。如 while 语句、do...while 语句和 for 语句。

(4) 函数。函数是实现某种特定功能的程序段,该程序段又是由顺序结构、选择结构和循环结构等所组成的。通过函数的使用,使程序段的功能相对集中,使程序的结构更具有层次性、更清晰。函数也称为模块结构。

2. 模块的特点

模块是结构化程序的基本组成单元,是数据说明、可执行语句等程序对象的集合。它是单独命名的而且可以通过名字来使用它。模块化就是把一个程序划分成若干个模块,每个模块完成一个子功能,通过模块的适当的组合而构成一个整体,完成指定的功能,满足问题的要求。因此,模块应具有以下特点:

(1) 模块独立。每个模块功能单一、明确,模块间的关系简单,结构清晰。

(2) 每个模块只有一个入口和一个出口。这样,当从顶部进入模块并且从底部退出来时,软件是便于理解和容易维护的。

(3) 模块规模适当。模块不能太大,也不能太小。模块太大,则功能复杂,可读性不好。模块太小,反而增加程序的复杂性。有人建议模块宜控制在 50 行~200 行之间。

(4) 模块能单独进行编译,模块可以嵌套,形成多层次结构。

3. 自顶向下的结构化程序设计方法

著名学者 Wirth 曾经说：“我们处理复杂问题最重要的办法是抽象。因此，对一个复杂问题不应该立刻用计算机指令、数字和逻辑符号来表示，而应该用较自然的抽象语句来表示，从而得出抽象程序。对抽象的数据进行某些特定的运算并用某些合适的记号来表示。对抽象程序做进一步的分解，并进入下一个抽象层次……，这样的求精化过程一直进行下去，直到程序能被计算机接受为止。这时的程序应该是用某种高级语言或机器指令书写的。”

对于一个较为复杂的问题，通常采用的是自顶向下逐步求精的方法。也就是先设计系统的整体结构，然后实现细节，即先抽象地从全局出发，按功能（或按任务）将问题逐步分解为几个大部分，然后将每个大部分再分解为相对简单的若干小部分，直到将问题分解为一个个很容易求解的小问题，如图 1-1 所示。

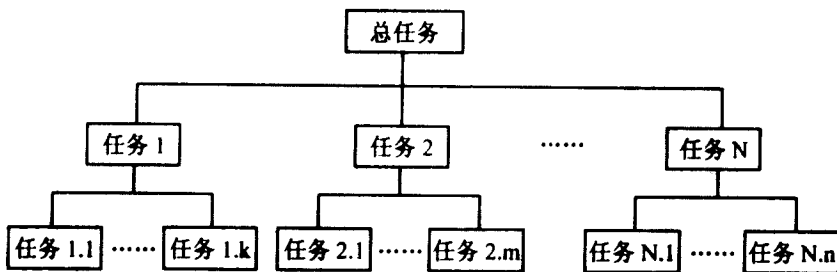


图 1-1 自顶向下细分示意图

自顶向下逐步求精的方法不仅用在系统的总体设计阶段，还可以用在系统的详细设计阶段，或者说是在软件开发的每一个阶段都有效的方法和手段。在系统的总体设计阶段，采用自顶向下逐步求精的方法，可以把一个复杂问题的解法分解和细化成一个由许多模块组成的层次结构的软件系统，如图 1-2 所示。在详细设计或编码阶段，采用自顶向下逐步求精的方法，可以把一个模块的功能逐步分解和细化为一系列具体的处理步骤或某种高级语言的语句。

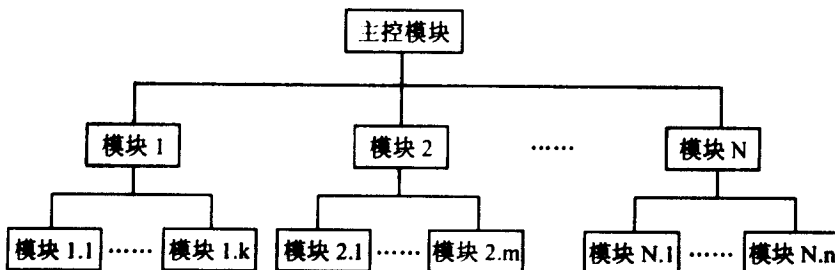


图 1-2 系统模块化结构图

使用结构化程序设计技术主要有下面几点好处：

(1) 自顶向下逐步求精的方法符合人们解决复杂问题的普遍规律，可以显著提高软件开发的成功率和生产率。

(2) 采用先全局后局部、先整体后细节、先抽象后具体的逐步求精过程开发出的程序层次清晰、增强可读性、便于理解解题思路。

(3) 使用单入口单出口的控制结构，使程序的表态结构和它的动态执行情况相对一致，便于错误的诊断与修改、提高开发的正确性、增强可读性和可理解性。

(4) 控制结构具有确定的逻辑模式，程序仅由很少几种语句构成，源程序清晰、易写、易读、易懂、易测试。

(5) 程序清晰和模块化使得软件开发中有大量的代码可以重用，减少劳动时间，提高了生产率。

第 2 章 C 语言实用程序设计技术

知识点:

- 图形处理函数及图形处理基本方法
- 菜单程序设计方法
- 屏幕图形的存取
- 简易动画设计技术
- I/O 接口输入/输出
- DOS 中断服务程序
- 驻留程序设计方法
- 汉字处理技术

本章导读:

本章主要综合程序设计中常用的一些工具和处理方法,为读者设计和开发综合实用程序提供必要的知识准备。

2.1 图形处理基础

当应用程序工作在图形界面下时,输出的效果可以丰富多彩,提高应用程序的界面友好性。通常计算机屏幕和图形功能的实现与计算机硬件紧密相连,而不同计算机硬件接口与性能又有着很大的差别。尽管 ANSI 标准的 C 语言没有定义屏幕与图形功能,但 Turbo C 提供了丰富的屏幕操作和图形功能函数,为用户开发图形程序提供了极大的方便。本节以 Turbo C 为例来简单说明图形处理的基本方法。

2.1.1 图形程序运行的条件

图形程序运行并显示图像直接与显示设备有关,图形程序必须控制驱动显示适配器进行显示。Turbo C 并没有向用户提供这种技术,但用户可以通过将相关的图形驱动程序连接到自己的图形程序中来实现。由于这些驱动程序一般由生产厂家提供,因此不同种类显示适配器的驱动程序也不同。但是,对于一般标准的显示适配器,如 CGA、EGA、VGA 等,其驱动程序已经在 Turbo C 系统上提供。

TC2 系统提供的显示适配器的图形驱动程序主要有: att.bgi、cga.bgi、egavga.bgi、herc.bgi、ibm8514.bgi、pc3270.bgi 和 bgiobj.exe 等。这些驱动程序一般在安装时存放在 TC 目录下。对于图形处理程序,若在所指路径下没有找到相应的显示驱动程序或没有对驱动

程序进行装入操作，则运行图形处理程序时就会出错，并显示出错信息：

BIG Error: Graphics not initialized (use "initgraph")

因此，图形程序运行的基本条件是：将相应的显示驱动程序装入内存运行。在C语言中，可以通过调用函数 `initgraph` 对图形系统进行初始化。

当正确地完成图形系统初始化后，生成的.EXE文件就可以在DOS下直接运行了。

2.1.2 图形显示器与适配器

一种图形显示器支持多种图形适配器，三种最常见的形式是彩色图形适配器（CGA）、增强图形适配器（EGA）和视频图形适配器（VGA）。对于这三种适配器，分别有多种不同的视频模式，每一种视频模式惟一地确定了图形显示中支持的最大颜色数和分辨率。表2-1给出了常用的图形适配器和图形模式。

表 2-1 图形适配器和图形模式

图形适配器		图形模式		颜色	分辨率
符号常数	数值	符号常数	数值		
CGA	1	CGAC0	0	4	320×200
		CGAC1	1	4	320×200
		CGAC2	2	4	320×200
		CGAC3	3	4	320×200
		CGAHI	4	2	640×200
EGA	3	EGALO	0	16	640×200
		EGAHI	1	16	640×350
VGA	9	VGALO	0	16	640×200
		VGAMED	1	16	640×350
		VGAHI	2	16	640×480

Turbo C 为用户提供了一个功能很强的图形接口，支持CGA、EGA、VGA等多种显示适配器的各种显示模式。Turbo C的图形接口包括图形库文件（`graphics.lib`）、图形头文件（`graphics.h`）和图形显示驱动程序（`*.BGI`）等。

提示：

CGA: Color Graphics Adapter, 彩色图形适配器。

VGA: Video Graphics Adapter, 视频图形适配器。

EGA: Enhanced Graphics Adapter, 增强型图形适配器。

在图形操作中可以使用什么样的颜色在不同的图形模式下是不一样的。在VGAHI模式下可以使用16种颜色，TC为它们分别定义了符号常量，各种颜色的数值及符号常量如表2-2所示。