

高等院校计算机教育系列教材

# C/C++ 程序设计教程

张莉 主编  
段清玲 陈雷 编著  
孟超英 主审

01



清华大学出版社

高等院校计算机教育系列教材

# C/C++程序设计教程

张 莉	主编
段清玲 陈 雷	编著
孟超英	主审

清华大学出版社

北 京

## 内 容 简 介

C语言具有简洁流畅、实用性强、代码编译效率高、可移植性强等优点，是一种通用语言，不仅适用于系统软件及应用软件的开发，也是现代程序设计的基础。要学习现代风格的编程技术，应该首先学会使用C程序设计语言。

本书主要内容包括程序设计的算法描述、C程序设计的结构特点、发展及应用、数据结构、表达式运算、程序流程控制、数组与函数、指针类型的各种操作、构造类型及应用、文件操作和C++程序设计等。内容完整详实，每一章配有思考练习题，非常适合于高校作为教材用书，也适合于各种培训。

本书作为新一轮教改用书，其内容的综合性、实用性和系统性也非常适合于编程爱好者及参加全国计算机等级(二级)考试人员作为自学参考书。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

### 图书在版编目(CIP)数据

C/C++程序设计教程/张莉主编；段清玲，陈雷编著；孟超英主审.—北京：清华大学出版社，2004  
ISBN 7-302-07929-3

(高等院校计算机教育系列教材)

I. C… II. ①张…②段…③陈…④孟… III. C语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2004)第001132号

出版者：清华大学出版社

<http://www.tup.com.cn>

社总机：010-62770175

地 址：北京清华大学学研大厦

邮 编：100084

客户服务：010-62776969

组稿编辑：彭欣

文稿编辑：刘颖

封面设计：陈刘源

印装者：清华大学印刷厂

发 行 者：新华书店总店北京发行所

开 本：185×260 印张：20.75 字数：483千字

版 次：2004年2月第1版 2004年2月第1次印刷

书 号：ISBN 7-302-07929-3/TP·5754

印 数：1~5000

定 价：26.00元

---

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770175-3103 或(010)62795704。

# 丛 书 序

人类进入新千年时，计算机科学已经具有一块极其活跃的、崇尚发展与创新的领地，并成为我们这一时代决定性的先驱技术。计算机是现代文化构成不可或缺的组成部分，是现代科学技术发展的先导，并且是世界经济巨量增长的根本引擎。同时，计算机技术的发展日新月异，它的快速发展对计算机科学与工程的教育产生了深刻的影响。因此，计算机教育在世界各国备受重视，计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

伴随着计算机新技术的不断涌现，已有技术几年内即变得陈旧。与此同时，计算机教育也被其所在的专业、文化和社会范围的改变影响着。计算机学科已变得更宽广、内容更丰富，其应用领域不断飞速增长。因此，无论在教学体系、教学内容还是教学方法、教学手段上都必须进行深化改革，与时俱进。CC2001 的出现，无疑是对计算机学科课程体系一个崭新的完整的引导。我们工科院校无论计算机专业还是非计算机专业的计算机科学与工程的教育应该紧密有机地与工科学生的培养目标相结合，注重知识、能力、素质教育三方面的综合教育，加强计算机系统的认知、分析、设计和应用能力，算法设计与分析能力，程序设计能力以及计算思维能力等方面的培养。

原化工部部属高校计算机教育协作组结合工程教育的特点，大力开展计算机教育协作与交流，十年来开展了务实的、全方位的、卓有成效的教学研讨及教研观摩等活动，极大地促进了交流并推动了各校计算机教育的发展。同时，协作组不断地扩展，吸收了许多其他领域的高校参加，共同为我国计算机高等教育事业的发展与完善进行广泛的交流探索。

目前参加这个协作组的主要高校有：

清华大学	南京大学	天津工业大学	中国农业大学
北京化工大学	南京工业大学	青岛科技大学	郑州大学
武汉化工学院	沈阳化工学院	南京师范大学	华南理工大学
河北行政学院	南京工程学院	淮海工学院	北京石油化工学院
江苏石油化工学院			

在清华大学出版社的大力支持下，本协作组 2001 年年会决定组织出版一套最新的计算机系列教材，第一期出版 14 部有关程序设计与软件应用方面的教材。它们是：《计算机导论》、《C 语言程序设计》、《Visual Basic 程序设计》、《Java 程序设计》、《面向对象程序设计——Visual C++》、《SQL Server 数据库原理及应用教程》、《C#编程及应用程序开发教程》、《组网技术与配置》、《数据库理论及应用基础》、《现代语音技术基础与应用》、《计算机图形学基础教程》、《Win32 汇编语言实用教程》、《C/C++程序设计教程》和《数据结构教程》等。

本系列教材依据 CC2001 框架，精心策划、准确定位，概念清晰，例题丰富，深入浅出，内容翔实，体系合理，重点突出，是一套面向高等学校计算机和非计算机专业学生的计算机基础与应用系列教材，也可供从事计算机应用和开发的各类人员学习使用。

本系列教材源于十几所全国重点大学和普通高等院校计算机教育的教学改革与实践，凝聚了工作在教学第一线的任课教师的教学经验与研究成果。我们期望本系列教材的出版，并在教学实践中不断完善与更新，为我国高校计算机教育事业做出新的贡献。

编委会  
2003 年 12 月

## 编委会名单

主编：朱群雄

编委：张 莉 闵华清 王晓峰

邵定宏 刘川来 彭四伟

刘 斌 刘新民 张彦锋

吕纪国 刘 焯 王相林

蔡莲红 孙正兴 冉林仓

彭 波

# 前 言

新世纪对信息技术人才掌握信息技术的综合能力要求越来越高，需求也越来越多。随着现代编程技术的广泛应用与发展，C/C++程序设计更为普及。学习和掌握 C/C++语言已成为现代软件开发人员的必备知识之一。

C 程序设计诞生于 20 世纪 90 年代后半叶，Unix 是世界上第一个易于移植的计算机操作系统。至今，C 语言程序设计一直广为普及、应用与发展，一直为专业系统开发人员和专业程序设计人员所青睐，并随着现代技术的发展，得到更好的完善。

使用 C 语言设计程序，编程结构简洁流畅、编译运行效率高。C 语言因其具有不同于其他语言的优点，具有很强的生命力，因此在快速发展的计算机技术领域，C 语言程序设计才能够持久地存在并得以丰富和发展。至今 C 语言程序设计仍是掌握各种 C 语言集成开发环境的重要基础。

C 语言程序设计不仅有高级语言的语义特点，还可以编写操作系统、编译程序、数据库软件系统等，还具有低级语言底层操作的能力，也可以方便地进行字节运算、字位运算、寄存器操作、取地址、设计中断、访问设备端口等。C 语言是现代技术人员不可或缺的软件系统开发与制作工具。随着计算机技术飞速的发展，C 语言本身也在不断地改进与发展，这种发展仍在继续，如现在广为使用的面向对象技术的 C++、可视化编程的 Visual C、网络编程语言等，但其基本语法规范与编程风格都没变。可以说掌握了 C 语言，就掌握了深入系统学习和应用计算机的钥匙，用户再学习理解其他现代编程技术也就方便、容易了许多。C++语言对 C 语言做了很多改进，例如增加了运算符、增加了类型的安全性、允许函数重载等。而 C++语言与 C 语言的本质区别是增加了面向对象的技术，如类、对象的封装性，基类、派生类的继承性，重载、动态联编多态性等。又如像 Sun Microsystems 公司 1995 年正式推出的新一代编程语言 Java，就是一种面向对象的跨平台、适合于网络计算模式的分布式语言，而 Java 程序设计就保留了许多广为熟知的与 C、C++同样的语法规范与结构风格。

熟练掌握 C 语言程序设计与掌握许多其他高级程序设计相比，还有一大优点，就是 C 语言程序设计可以很好地发挥人的创新与创造思维，达到“能想到的就能实现”的境地。

当然，学习和掌握 C 语言并不像其他高级语言那样容易。特别是在掌握 C 语言程序设计精华或在深入、系统应用等方面，还需涉及到一些计算机专业基础知识，如操作系统、数据结构、数据库系统、软件工程等。有些内容还需要大量的实践积累，才能提高综合应用能力，充分发挥自己的创新思维，让创造付诸于实际应用与开发。

本书针对现代教育教学改革理念，在提高教学效率的同时，有效提高学生综合实践的能力。本书集结了许多多年以来在软件开发与 C/C++程序设计教学方面富有实践经验的一线教师，根据现代高校教学改革特有的情况，探讨了现代计算机教学的规律，收集分析了大量的教学文献，并基于实际应用编写了本书。本书内容翔实，概念清晰，通过实例分析，

可以引导读者尽快掌握 C/C++程序设计的实际应用技巧。本书作为教学改革用书，内容编排比较紧凑，可以帮助读者系统全面地了解 C 程序设计的重点和要点，并引导实际编程与应用。

整个教程分为 15 章，其中第 1、2、4、13 章由张莉教授负责编写；第 3 章和第 6 章由孟超英教授负责编写；第 5 章和第 7 章由冀荣华讲师负责编写；第 9 章和第 10 章由段清玲副教授负责编写；第 8 章由郑立华讲师负责编写；第 11 章由陈雷副教授负责编写；第 12 章由李琼飞讲师负责编写；第 14 章和第 15 章由孙龙清教授负责编写。全书由张莉教授统一组编修改定稿。

本书作为教学实践用书，内容编排上注重基础性和实用性。希望各界读者广泛交流，提出宝贵意见。

本书基于教改教学，侧重于培养学生提高综合实际应用技能，也很适于自学使用。参加本书编写审校的人员还有张筠、杨丽丽、吕春利、马钦等教师和技术人员。另外，叶海建教授也在各方面给予了大量的帮助与支持，均在此致以谢意。由于本书源于实践，在内容取材与编写方面力求要点突出、结构紧凑，易读易懂。

本书在组织时间上比较仓促，难免有不足之处，还需要在实际教学中进一步提炼与完善。诚望广大读者提出宝贵的意见，共同探索新世纪 IT 教育的有效方法，以尽快适应现代技术对人才培养目标的需要。

编者

2003 年 12 月

# 目 录

<b>第 1 章 计算机程序设计及算法</b> ..... 1	
1.1 程序设计..... 1	
1.1.1 程序设计语言..... 1	
1.1.2 程序设计过程..... 3	
1.2 程序设计的算法..... 4	
1.3 计算机算法的表示..... 4	
1.3.1 自然语言描述..... 5	
1.3.2 程序流程图描述..... 5	
1.3.3 N-S 图描述..... 7	
1.4 用程序设计语言描述..... 9	
1.5 算法举例..... 12	
1.6 习题..... 16	
<b>第 2 章 C 语言程序设计概述</b> ..... 17	
2.1 C 语言概述..... 17	
2.1.1 C 语言的发展..... 17	
2.1.2 C 语言的特点..... 18	
2.2 Turbo C 的集成环境..... 20	
2.2.1 Turbo C 2.0 的特点与配置要求..... 20	
2.2.2 Turbo C 2.0 的安装与启动..... 20	
2.2.3 Turbo C 的快捷键..... 23	
2.3 C 语言的程序结构..... 24	
2.3.1 C 语言程序的组成..... 24	
2.3.2 C 语言的标识符..... 26	
2.3.3 C 语言的关键字..... 26	
2.4 习题..... 27	
<b>第 3 章 数据类型、运算符和表达式</b> ..... 28	
3.1 常量和变量..... 28	
3.1.1 常量..... 29	
3.1.2 变量..... 31	
3.2 整型数据类型..... 32	
3.2.1 整型变量的分类..... 32	
3.2.2 整型变量的定义..... 34	
3.2.3 整型常量的表示方法..... 35	
3.2.4 整型常量的分类..... 35	
3.2.5 整型数据的应用..... 36	
3.3 实型数据类型..... 36	
3.3.1 实型常量的表示方法..... 36	
3.3.2 实型变量..... 37	
3.3.3 实型数据的应用..... 39	
3.4 字符型数据类型..... 39	
3.4.1 字符常量..... 39	
3.4.2 字符变量..... 40	
3.4.3 字符数据在内存中的存储形式及使用方法..... 40	
3.4.4 字符串常量..... 41	
3.4.5 字符数据的应用..... 42	
3.5 不同类型数据间的混合运算..... 42	
3.6 算术运算符和算术表达式..... 43	
3.6.1 运算符简介..... 43	
3.6.2 算术运算符和算术表达式..... 44	
3.7 赋值运算符和赋值表达式..... 47	
3.8 逗号运算符和逗号表达式..... 49	
3.9 变量赋初值..... 50	
3.10 本章小结..... 50	
3.11 习题..... 51	
<b>第 4 章 顺序结构程序设计</b> ..... 53	
4.1 顺序结构概述..... 53	
4.2 基本顺序结构语句..... 54	
4.2.1 表达式和表达式语句..... 55	
4.2.2 基本顺序命令..... 56	
4.3 基本输入/输出..... 58	
4.3.1 字符输入函数..... 58	
4.3.2 字符输出函数..... 59	



4.3.3 格式化输入/输出函数 .....	60	7.3.1 字符数组 .....	119
4.4 习题 .....	64	7.3.2 字符串 .....	121
<b>第 5 章 流程控制——选择结构</b> .....	67	7.4 字符串处理函数 .....	123
5.1 各种分支结构 .....	67	7.5 字符数组的应用举例 .....	127
5.2 if 语句 .....	67	7.6 本章小结 .....	129
5.2.1 if 语句 .....	68	7.7 习题 .....	130
5.2.2 if-else 语句 .....	69	<b>第 8 章 函数与变量</b> .....	136
5.2.3 else-if 语句 .....	71	8.1 函数 .....	136
5.2.4 if 语句的嵌套 .....	74	8.1.1 函数的定义 .....	136
5.3 条件运算符 .....	77	8.1.2 函数的参数和返回值 .....	140
5.4 switch-case 语句 .....	79	8.1.3 函数的声明和调用 .....	141
5.5 本章小结 .....	82	8.1.4 函数的递归调用 .....	150
5.6 习题 .....	82	8.1.5 外部函数与内部函数 .....	153
<b>第 6 章 循环控制</b> .....	88	8.2 变量的作用域和存储类型 .....	156
6.1 for 语句 .....	88	8.2.1 变量的作用域 .....	156
6.2 while 语句 .....	91	8.2.2 变量的存储类型 .....	158
6.3 do-while 语句 .....	93	8.3 本章小结 .....	164
6.4 循环的嵌套 .....	95	8.4 习题 .....	164
6.5 goto 语句、break 语句和 continue 语句 .....	96	<b>第 9 章 编译预处理</b> .....	169
6.5.1 goto 语句 .....	96	9.1 编译预处理简介 .....	169
6.5.2 break 语句 .....	97	9.2 宏定义和宏替换 .....	169
6.5.3 continue 语句 .....	98	9.2.1 符号常量的宏定义 .....	170
6.6 循环结构比较 .....	99	9.2.2 带参数的宏定义 .....	171
6.7 本章小结 .....	100	9.3 文件包含 .....	174
6.8 习题 .....	100	9.4 条件编译 .....	175
<b>第 7 章 数组与字符串</b> .....	105	9.5 本章小结 .....	178
7.1 一维数组 .....	105	9.6 习题 .....	178
7.1.1 数组的基本概念 .....	105	<b>第 10 章 指针</b> .....	181
7.1.2 一维数组的定义和引用 .....	106	10.1 指针的基本概念 .....	181
7.1.3 一维数组的初始化 .....	108	10.2 指针变量和指针运算符 .....	183
7.1.4 一维数组应用举例 .....	109	10.2.1 指针变量的定义 .....	183
7.2 多维数组 .....	112	10.2.2 指针变量的引用和 初始化 .....	183
7.2.1 二维数组的定义和引用 .....	113	10.2.3 指针的运算 .....	185
7.2.2 二维数组的初始化 .....	115	10.3 指针和函数参数 .....	187
7.2.3 三维数组 .....	117	10.4 指针与数组 .....	189
7.3 字符与数组 .....	118	10.4.1 用指针访问数组元素 .....	190

10.4.2 指针与多维数组.....	193	<b>第 12 章 枚举与位操作</b> .....	237
10.4.3 用指针访问字符串.....	195	12.1 枚举.....	237
10.4.4 指针数组.....	198	12.2 类型定义.....	239
10.4.5 指向指针的指针.....	199	12.3 位运算.....	242
10.5 指针与函数.....	200	12.3.1 位运算符.....	242
10.5.1 函数指针.....	200	12.3.2 位域(位段).....	247
10.5.2 函数指针作为函数 参数.....	202	12.3.3 位运算应用.....	249
10.5.3 返回指针的函数.....	203	12.4 本章小结.....	250
10.5.4 main 函数的参数.....	204	12.5 习题.....	251
10.6 本章小结.....	205	<b>第 13 章 文件</b> .....	253
10.7 习题.....	205	13.1 文件概述.....	253
<b>第 11 章 结构体与共用体</b> .....	209	13.1.1 文件的概念与文件 结构.....	253
11.1 结构体的定义与引用.....	209	13.1.2 文件系统的缓冲性.....	254
11.1.1 结构体的定义.....	210	13.2 标准文件输入/输出.....	254
11.1.2 结构体的引用.....	212	13.3 缓冲型文件的输入/输出.....	255
11.2 结构体的初始化.....	213	13.3.1 文件类型指针.....	255
11.3 结构体数组.....	215	13.3.2 缓冲型文件输入/输出 函数.....	255
11.4 指向结构体的指针.....	216	13.3.3 文件的打开与关闭.....	256
11.4.1 结构体指针的概念.....	216	13.3.4 文件的读写.....	259
11.4.2 结构体指针的使用.....	217	13.3.5 文件的定位.....	263
11.4.3 指向结构体类型数组的 指针的使用.....	218	13.3.6 出错检测.....	265
11.5 结构与函数.....	220	13.3.7 其他缓冲型文件函数.....	266
11.6 结构体指针与链表.....	222	13.4 本章小结.....	267
11.6.1 什么是链表.....	222	13.5 习题.....	268
11.6.2 链表的建立和输出.....	222	<b>第 14 章 C++的面向对象程序设计</b> .....	270
11.6.3 链表的插入.....	225	14.1 C++的特点.....	270
11.6.4 链表的删除.....	226	14.2 由 C 向 C++过渡.....	271
11.7 共用体.....	228	14.2.1 任何函数调用之前必须 有函数声明或定义.....	271
11.7.1 共用体的定义.....	228	14.2.2 函数声明时形式参数 不能省略.....	271
11.7.2 结构体与共用体的 区别.....	229	14.2.3 关于自定义数据类型 变量的定义.....	272
11.7.3 共用体的引用.....	229	14.2.4 注释行.....	272
11.8 综合举例: 学生管理系统.....	231	14.2.5 声明语句.....	272
11.9 本章小结.....	234		
11.10 习题.....	234		

14.2.6 C++的输入输出.....	273	14.6 函数的多态性.....	293
14.2.7 作用域运算符.....	274	14.6.1 成员函数的重载.....	294
14.2.8 内联函数.....	274	14.6.2 虚拟函数(虚函数).....	295
14.2.9 带默认形参值的函数.....	275	14.7 本章小结.....	298
14.2.10 函数的重载.....	277	14.8 习题.....	299
14.2.11 new 和 delete 运算符.....	278	<b>第 15 章 C++实例分析</b> .....	<b>300</b>
14.3 面向对象的方法.....	280	15.1 问题提出.....	300
14.3.1 面向对象的语言.....	280	15.2 类的设计.....	300
14.3.2 面向对象方法.....	280	15.3 程序及说明.....	300
14.4 类与对象.....	281	15.4 运行结果.....	304
14.4.1 类的定义.....	281	<b>附录 A 常用字符与 ASCII 码对照表</b> .....	<b>306</b>
14.4.2 类的成员函数.....	282	<b>附录 B 各章习题参考答案</b> .....	<b>308</b>
14.4.3 构造函数和析构函数.....	285	<b>附录 C 参考文献</b> .....	<b>317</b>
14.4.4 对象.....	287		
14.5 派生类.....	291		
14.5.1 派生类的声明.....	291		
14.5.2 多重继承.....	292		

# 第 1 章 计算机程序设计与算法

以计算机技术、网络技术与通信技术为基础的 IT 信息技术广泛地普及与应用,推动了知识经济的兴起。知识经济也是信息经济,是以信息产业为标志、以知识为基础的经济。社会要跟上信息时代的步伐,使整个经济持续稳定地发展,就要发展知识经济,就必须加强和提高现代人才的信息化教育;个人熟练掌握计算机程序设计技术与算法,是开发利用信息技术最基本的技能之一。实际上学习计算机的知识并不仅仅是学习操作和一般应用程序的使用,计算机作为一种工具,在学习的同时,若能熟练掌握一到两种程序设计语言及现代编程设计方法,则可以把我们带到一个全新的知识领域,使我们真正接触到更为广阔的信息科技世界。

本章主要内容:

- 程序设计语言
- 程序设计过程
- 程序设计的算法
- 自然语言描述计算机算法
- 程序流程图描述计算机算法
- N-S 图描述计算机算法
- 用程序设计语言描述计算机算法

## 1.1 程序设计

学习程序设计并不是简单地学习计算机语法规则或程序设计语言的本身,而是要学会怎么用计算机程序设计语言解决实际问题、提高工作效率和工作质量。计算机技术应用领域博大而广泛,谁也不可能完全学会计算机技术的方方面面,计算机技术的发展日新月异,学无止境。

### 1.1.1 程序设计语言

多年以来国内外教育界专家一直认为计算机教育必需要学习程序设计语言和设计方法,特别是高等教育,要加强信息技术的教育,更应加强力度培养学生学好程序设计语言及现代编程方法。这不仅可以培养学生运用程序设计的算法来解决实际问题的应用能力,提高专业技术拓展能力,也可以提高信息技术综合应用技能。如今我国许多中小学在具备了相应的师资与教学实验条件后,也相继开设了不同层次的计算机教育与编程语言课,主要以 Basic 语言或 Visual Basic 为主,但地区差异很大,有的中学计算机设备和实验条件相当

好，还接入了互联网，实现了远程教学；而更多的边远地区的学生却很少触及到现代信息设施，更不要说掌握应用技能了。我国高校普遍开设的计算机基础教育课程系列，即在相对有保障的教学计划和教学实验条件下，使非计算机专业的学生系统有效地掌握了现代信息技术的基础理论和应用技能，成为培养现代人才的重要环节，甚至造就出一大批现代 IT 精英，不仅提高了个人的就业实力，也提高了国际经济活动的行业竞争能力。目前，国家教育部仍在进一步加强高等教育中现代信息技术环节的教育。

实际上能实现各种算法、被称之为语言的集成环境很多，可以说高校公共基础教育教什么样的语言就其基本教学目标而言并无多大差异，语言实现算法的本质结构都是相似的，如果有区别就是应用功能上的不同、应用平台上的不同和函数库的不同。要是我们对语言进行分类，可以从许多方面来分：可以按对机器的依赖程度分类、按程序设计方法学分类、按计算方法分类以及按应用领域分类等。例如我们可以从应用领域的角度分类。

- (1) 科学计算语言 用于科学计算，基础是数学模型，过程描述的是数值计算，如 FORTRAN 语言；
- (2) 系统开发语言 用于编写编译程序、操作系统、数据库管理系统 DBMS 等，如 C 语言；
- (3) 实时处理语言 及时响应环境信息，可以根据外部信号对不同的程序段进行并发控制执行；
- (4) 商用语言 主要用于商业处理、经济管理，基础为自然语言模型；
- (5) 人工智能描述语言 模拟人的思维推理过程，实现智能化控制等；
- (6) 模拟建模语言 用于模拟实现客观事物的发展与变化过程，以提前预测未来发展的结果；
- (7) 网络编程语言 在网络技术基础上进入深层次的应用研究与开发的语言，如 Java 是一种新型跨平台分布式程序设计语言，Delphi 适于网络化环境的编程等，不再列举。

就技术性、基础性和实用性而言，选用哪种程序设计语言系统学习最为合适呢？可以说在各种现代编程语言尽显魅力的今天，仍当属 C 程序设计语言最具有代表意义，这也是国内外高校计算机教育中经久不衰的原因之一。C 语言编程流畅、编译效率高、应用广泛，也是现代网络编程如 Java、面向对象可视化编程 Visual C++ 等程序设计的基础。我国高校 1993 年前后逐步引入了 C 语言程序设计，由于 C 语言最早产生于 Unix 操作系统平台，后来普及到微型计算机上，由于其编码效率较高，能进行底层操作，如机器的中断地址、设备端口、寄存器、字位操作等，还可以处理字符和图形、编写界面；它使用方便，是开发系统软件的良好选择。对于学生来说，如果能掌握一些硬件的基本知识，熟练掌握 C 语言后，将会成为就业的优势之一。

如今，学好计算机技术已经成为人们完善自我、提高工作技能的重要需求，是人们现代生活中的一部分，因为相当多的职业都离不开计算机，而计算机的发展与运用正是现代科学技术发展的重要标志。充分利用和掌握，才能达到期望的职业技能应用水平。

计算机可以代替人从事重复性的劳动，并能完成人所不能做到的事情，有人误以为计算机就是操作使用的工具，计算机工作的每一步都要靠人手工操作完成；而另一些人对计算机的作用估计过高，认为只要有了计算机，一切事情都可以自动办到。其实，计算机是

按照人为设计的步骤, 编制好程序, 事先输入计算机, 执行后才能自动连续并能重复运行, 从而精确、快速、规范地代替人进行复杂繁琐的工作。计算机是集人类智慧设计和制造的, 因此计算机不可能完全代替人, 仍然需要人来利用计算机进行开创性的劳动, 因此, 有必要利用算法与程序实现人们要做的各种事情, 借助计算机完成更为复杂的工作。

## 1.1.2 程序设计过程

### 1. 程序实现过程

程序设计是指使用一种计算机语言为实现解决实际问题的算法去设计编写计算机程序的过程。计算机语言是人与计算机进行交流的媒介, 通过语言编写的程序, 计算机就会准确地按程序步骤执行操作, 计算机解决实际问题的过程如图 1.1 所示。

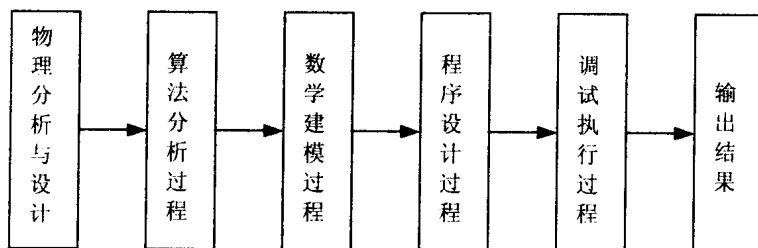


图 1.1 计算机解决实际问题的过程

其中物理过程是算法设计的基础, 需建设和确定实现算法的运行环境, 需要大量的实地调研和分析工作; 算法分析及数学建模过程是实现具体算法的重要步骤, 应有必要的专业基础理论与专业背景, 还应具备实际经验, 因此常常由具有专业技术背景的人员与程序员共同协作完成。接下来才是程序员的程序设计与调试工作, 其中的关键是算法分析与建模过程, 重点是程序实现。

### 2. 程序设计的过程

程序设计大致包含以下几个步骤:

#### (1) 分析问题

程序设计首先要对问题进行分析, 明白我们要做什么, 确定要使用的数学模型。

#### (2) 确定算法

确定算法即确定解决问题时要执行的一系列步骤。

#### (3) 算法描述

算法描述就是使用计算机语言对算法予以描述。

#### (4) 确定程序设计语言

由于不同的计算机程序设计语言有不同的特点, 根据实际情况与需要选定好程序设计语言后, 就可以用该语言编程以实现算法。

#### (5) 调试和运行程序

把编好的源代码程序输入计算机, 利用程序设计语言集成开发环境或工具进行调试和运行。

## 1.2 程序设计算法

学习计算机语言程序设计，语法是编写程序的文法规则，不能有误，否则计算机语言编译或解释系统就会检测出程序有语法错误，用户的计算机程序就不会被“翻译”成机器语言正确执行，因此应掌握准确的语法规范。而解决问题的算法可以自由灵活地优化设计，因此同一个问题的算法可以有多种，可编的程序也不是惟一的。所以学习程序设计，熟练掌握语法规范和语言结构是必要的，而算法实现是核心，综上所述，计算机程序可以表示为：

程序=算法+数据结构

其中，

- (1) 数据结构：对数据的描述，包括对数据类型的描述和对数据组织形式的描述定义。
- (2) 算法：对操作过程的描述，即操作步骤的描述。

如果考虑现代编程的工程化与多样性，也可以这样表示：

程序=数据结构+算法+(程序设计方法+编程工具+语言环境)

其中算法是关键，是实现程序设计的依据和基础，算法分析做得完整和精细，才能有完整的程序设计，才可能对程序进行优化，所以掌握算法至关重要。

## 1.3 计算机算法的表示

人们在实际生活和工作中，无论做什么事情，都需要有计划、按步骤去完成，如果你计划去看一场电影或听一场音乐会，就需看海报，完成这项活动需要按指定的方式到指定的地点付款、购票，再按规定的时间到规定的地点，验票进门找到自己的座位，接下来看电影或听音乐会，完成计划和步骤。实施这件事情的整个过程是按时间、按步骤、顺序进行的，直至完成，这仅仅是人们日常生活所熟悉的许许多多事情中的一项，这个过程实施的步骤就是“算法”，只是人们习以为常，从没有去细究其思维支配的过程，也就不必细细描述。

要让计算机去做一件事情，就必须把如何做这件事情分拆成许多小步骤，每一步正好由计算机语言的命令来完成，计算机按着顺序执行这一系列小步骤，最后执行完一系列命令，把所要做的事情做完。

描述这些步骤的方法有自然语言、流程图和 N-S 图等，描述每一步骤的计算机命令时选用的就是计算机语言。把计算机每一条命令对应的一系列步骤组成一个规则的整体，就是计算机程序。计算机程序设计“算法”就是利用计算机语言，让计算机完成解决问题的步骤。那么假使解决问题的一项活动还穿插有其他活动，则需要周密计划、严谨安排，确定先做什么后做什么，再做什么，才可能有效完成任务，对应的计算机语言编程则是流程

控制问题。

顺序结构是程序设计最基本的结构，由顺序命令组成，程序执行流程是按顺序执行方式运行的。条件分支结构根据不同的条件执行不同的命令。循环结构根据给定的条件反复执行某一段程序，并不断修改给定的条件，最终结束循环。

### 1.3.1 自然语言描述

用自然语言描述算法是非常直接的一种表示方法，易于表达、易于理解，所以自然语言描述算法最大的优点是便于人们之间直接交流，其最大的缺点也产生于人们之间的交流，表现在自然语言表述不够严谨和理解不一致时，容易产生歧义性。

例如，“学期末获优秀学生奖的条件是：考核达标、平均成绩 90 分以上或从未迟到早退者，即可参加评奖。”

听起来很容易理解，但理解方式并不是惟一的。

理解 1：考核达标、平均成绩 90 分以上，即可参加评奖。

理解 2：考核达标从未迟到早退者，即可参加评奖。

理解 3：从未迟到早退者，即可参加评奖。

所以自然语言表述不能有效合理地表达算法，显然也不能利用计算机有效地进行程序设计和实现算法。

### 1.3.2 程序流程图描述

用自然语言描述程序执行步骤，如果是顺序结构还比较容易，但是稍微复杂一点的程序流程，例如条件判断分支结构或循环结构等，就不是很方便。因此，美国国家标准化协会 ANSI(American National Standard Institute)规定了一些常用的流程图符号，表示程序的执行步骤与控制流向，这就是程序流程图，如图 1.2 所示。

使用程序流程图表示程序算法，流程清晰，易看易懂，用这些符号描述程序的执行步骤是国际通用表示方法，在程序设计中普遍使用。

例如，用程序流程图表示这样一个算法：输入两个数，如果这两数之和大于 0，则输出结果；如果这两数之和小于 0，则重新输入两个数，如果两数之和等于 0，则结束。程序流程图如图 1.3 所示。

虽然用程序流程图表示算法，程序步骤和过程表达清楚、简明易看、容易理解，但是要表示复杂一些的算法就会显得凌乱繁杂，如图 1.4 所示。

由于这个原因，人们意识到应该规定几种基本的算法设计结构，然后按照一定的规范组合成整体的算法结构。这些基本结构可以组成各种新的基本结构，然后按顺序搭接构成整个程序的算法结构，使程序设计质量大为提高。

1966 年，Bohra 和 Jacopini 提出了程序设计的 3 种基本结构，即顺序结构、条件判断分支结构和循环控制结构，至今仍是各种计算机程序设计语言支持的程序流程控制基本结构。3 种程序控制结构表示如图 1.5 所示。



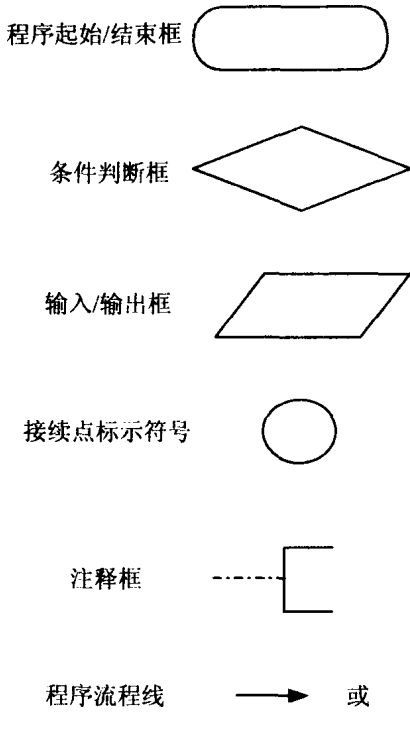


图 1.2 程序流程图基本符号

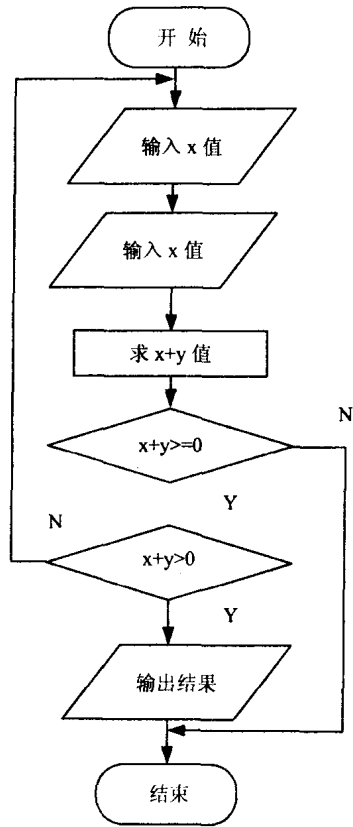


图 1.3 算法程序流程图

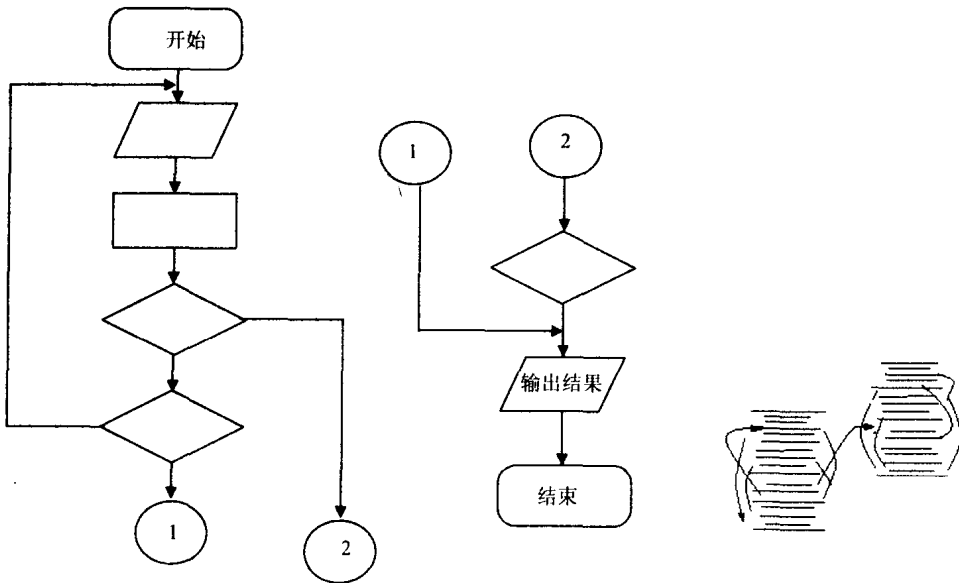


图 1.4 表示复杂的算法凌乱繁杂

实际上，使用上述 3 种基本结构很易组成任何一种复杂的程序算法。很明显，以上 3 种基本结构都是一个入口一个出口，用这 3 种基本结构组成的程序算法都属于结构化程序