新编21世纪高等院校计算机系列规划教材

C语言程序设计

北京希望电子出版社　总策划
张建伟　李秀芹　主　编
甘　勇　金保华　副主编

科学出版社
www.sciencep.com

新编21世纪高等院校计算机系列规划教材

C语言程序设计

北京希望电子出版社 总策划
张建伟 李秀芹 主 编
甘 勇 金保华 副主编

科学出版社
www.sciencep.com

内 容 简 介

程序设计技术和程序设计语言是大学计算机专业及相关专业开设的计算机程序设计的重要课程，其主要任务是培养学生的逻辑思维能力、抽象能力和基本的程序设计能力。本书从结构化程序设计技术出发，以C程序设计语言为载体，以Windows下的Visual C++ 6.0为程序调试和运行平台，通过对典型实例的算法描述以及相应C语言代码的描述，展现了在程序设计过程中如何对问题进行分析、如何组织数据和如何描述解决问题的方法，揭示了在计算机应用过程中如何将方法和编码相联系的具体程序设计过程，进而向读者介绍结构化程序设计的基本概念、基本技术和方法。

本书适合作为高等院校理工类各专业本、专科作为程序设计技术、程序设计语言的课程教材，也可供程序开发人员学习参考。

与本书配套的还有一本《C语言程序设计实验指导与习题解答》。

需要本书或技术支持的读者，请与北京清河6号信箱（邮编：100085）发行部联系，电话：010-82702660，62978181（总机）传真：010-82702698，E-mail：tbd@bhp.com.cn。

图书在版编目（CIP）数据

C语言程序设计 / 张建伟，李秀芹主编.—北京：科学出版社，2007.1

（新编21世纪高等院校计算机系列规划教材）

ISBN 978-7-03-018460-3

Ⅰ. C… Ⅱ. ①张… ②李… 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（2007）第008989号

责任编辑：王超辉 / 责任校对：马 君

责任印刷：媛 明 / 封面设计：梁运丽

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

北京媛明印刷厂

科学出版社发行 各地新华书店经销

*

2007年1月第 一 版 开本：787×1092 1/16

2007年1月第一次印刷 印张：19 3/4

印数：1—5000 字数：452 010

定价：28.00元

新编21世纪高等院校计算机系列规划教材编委会

总　序

一本好书，是人生前进的阶梯；一套好教材，就是教学成功的保证。为满足培养应用型人才的需要，我们成立了本编委会。在明确高等院校应用型人才培养模式、培养目标、教学内容和课程体系的框架下，我们组织编写了本套规划教材。

为了使本套教材能够达成目标，编委会做了大量的前期调研工作，在广泛了解各高等院校的教学现状、学生水平、培养目标的情况下，认真探讨了课程设置，研究了课程体系。为了编写出符合教学需求的好教材，我们除了聘请一批计算机知名专家、教授作为本套教材的主审和编委外，还组织了一批具备较高的学术水平、丰富的教学经验、较强的工程实践能力的学术带头人和骨干教师来承担具体编写工作，从而编写出特色鲜明、适用性强的教材，以真正满足目前高等院校应用型人才培养的需要。教材编写采用整体规划、分步实施、在实践中检验提高的方式，分期分批地启动编写计划。编写大纲以及教材编写方式的确定均经过编委会多次认真讨论，以确保该套教材的高质量和实用性。

本套规划教材的主要特点是：

（1）以服务教学为最高宗旨，认真做好教学内容的取舍、教学方法的选取、教学成果的检验工作。本套教材在教学过程中的有益反馈，都将及时体现在后续版本。

（2）面向应用型高等院校，在保证学科体系完整的基础上把握好理论的深度和难度。注重理论知识与实践相结合，使学生通过实践深化对理论的理解，学会并掌握理论方法的实际运用。从而较好地培养学生的专业技能和实施工程的实用技术能力。

（3）教材在内容编排上，力求由浅入深，循序渐进；举一反三，突出重点；语言简练，通俗易懂。采用模块化结构，兼顾不同层次的需求，在具体授课时可根据具体教学计划适当取舍内容。

（4）教材采用“任务驱动”的编写方式，以实际问题引出相关原理和概念，在讲述实例的过程中将本章的知识点融入，通过分析归纳，介绍解决工程实际问题的思想和方法，同时，引入案例教学和启发式教学方法，便于激发学习兴趣。

（5）在教材中加大实训部分的比重，使学生能比较熟练地应用计算机知识和技术解决实际问题，既注重培养学生分析问题的能力，也注重培养学生解决问题的能力。

（6）大部分教材配有电子教案，从而更好地服务教学。

为编写本套教材，作者们付出了艰辛的劳动，编委会的各位专家进行了悉心的指导和认真的审定。书中参考、借鉴了国内外同类的优秀教材和专著，在此一并表示感谢。

我们衷心希望更多的优秀教师参与到教材建设中来，真诚希望广大教师、学生与读者朋友在使用本丛书过程中提出宝贵意见和建议。

若有投稿或建议，请发电子邮件到 textbook@bhp.com.cn。谢谢！

新编 21 世纪高等院校计算机系列规划教材编委会

前　言

C 语言是应用最为广泛的一种高级程序设计语言，以它独到的优势和特点赢得了编程人员的青睐和信任。C 语言功能丰富，表达能力强，使用灵活方便，程序效率高，是结构化的程序设计语言。C 语言具有低级语言的许多特点，可直接处理字符，进行位运算和指针运算等。因此，C 语言具有很强的实用性，既可用来编写应用软件，也适合于编写系统软件。现在，近乎所有高等院校都将“C 语言程序设计”作为计算机专业必修课程和非计算机专业首选程序设计语言课程。C 语言程序设计也是全国和各省的计算机等级考试的重要考试内容。

随着计算机科学与技术的飞速发展，计算机的应用已经渗透到国民经济与人们生活的各个角落，正在日益改变着传统的人类工作方式和生活方式，在我国高等教育逐步实现大众化以后，越来越多的高等院校面向国民经济的第一线，培养各类高级应用型专门人才。基于以上的高校办学定位，着手本教材的编写。

总体思路是：切实把握计算机软件开发这门学科的实际情况，从实际应用出发，以实践带动理论知识的学习，也就是说把程序设计的学习贯穿到编写程序的过程中。达到既满足学习程序设计技术知识、掌握编程技巧、备考和应考之目的，又能培养学生程序开发设计的能力和解决实际问题的能力。

在内容安排上分为两个部分，第一部分由第 1 章组成，主要介绍程序设计技术的相关知识，包括算法、数据结构、程序、程序设计语言、结构化程序设计和面向对象的程序设计方法与技术。使读者对程序设计技术的相关概念、理论有一个整体的了解。第二部分由第 2 章～第 10 章组成，介绍 C 语言的语法规则及编程技巧，该部分内容安排既考虑全国计算机等级二级考试的要求，同时又考虑读者进一步深入学习的要求，在每章最后都安排有综合应用实例。教师可以根据学生的实际情况进行选择。

具体到每一章节，在阐述过程中，肯定会涉及到本章内容以外的知识，在这里应该简单的解释，并且指出将在哪一部分作讲解。因为程序设计语言的特点是，在严整的体系结构上，又有着独体的图状联系，也就是各个部分的语法现象互相密切联系，到了无法严格区分的地步。所以，整体上可以把程序设计语言划分为几个独立的部分，但是在具体讲解过程中又互相密切关联。在现行的各种教程中，读者往往在具体的某一章节中碰到没有见过的语法现象，鉴于此种情况，在本教程中，要把这些脉络完整的展现出来，让读者能够知道哪些知识是本章内的内容，哪些知识不是本章的内容，但是与本章相关的内容密切相关。进而让读者知道自己学到了些什么。以避免一看到很多没有见过的知识便产生胆怯的感觉，以帮助读者树立继续学习的信心。

为适应技术和应用发展的需要，本书完全以 Visual C++ 6.0 作为程序调试平台，方便用户进行程序源代码的编辑，同时为进一步学习 C++奠定基础。

本书由张建伟、李秀芹任主编，甘勇、金保华副主编，参加本书编写的有：皇甫中民、卢冰、姚云星。其中第 1、9 章和附录由张建伟编写，第 2 章由甘勇编写，第 3、4 章由皇甫中民、甘勇编写，第 5 章由李秀芹、卢冰编写，第 6 章由姚云星编写，第 7 章由卢冰编写，第 8、10 章由金保华、姚云星编写；书中源代码由贺磊和蔡增玉负责调试，全书由张建伟统稿并定稿，由甘勇教授主审。在本书的编写和出版过程中得到了郑州轻工业学院教务处、科学出版社的大力支持和帮助，在此由衷地向他们表示感谢!

本书除了可用作高等院校本、专科学生的教材外，兼顾一般读者，可作为从事计算机应用开发人员在学习程序设计语言和程序设计技术时作为参考。

由于编者水平有限，本书难免有错误之处，恳请广大读者不吝赐教。

编　者

目　录

第 1 章　程序设计技术概述.....1
1.1　程序设计技术概要.....1
1.1.1　算法.....1
1.1.2　数据结构.....4
1.1.3　程序和程序设计语言.....5
1.2　结构化的程序设计方法与技术.....9
1.2.1　结构化方法的核心问题.....9
1.2.2　结构化设计.....9
1.2.3　结构化实现.....11
1.2.4　结构化方法的优点及问题.....12
1.3　面向对象的程序设计方法与技术.....12
1.3.1　面向对象方法的产生及要点.....12
1.3.2　面向对象的基本概念.....14
1.3.3　面向对象的软件开发过程.....15
1.3.4　面向对象方法的特点.....18
1.4　C 语言的产生及特点.....19
1.5　Visual C++ 6.0 环境运行的 C 程序示例.....22
1.5.1　Visual C++ 6.0 集成开发环境简介.....22
1.5.2　编辑、编译、运行 C++程序.....23
1.5.3　C/C++程序的调试.....27
1.5.4　如何在 Visual C++ 6.0 环境下编辑、运行 Turbo C 程序.....28
习题.....30
第 2 章　数据类型、运算符和表达式.....31
2.1　数据类型.....31
2.1.1　C 语言的数据类型.....31
2.1.2　数值型数据的表示与存储形式.....32
2.1.3　字符型数据的表示与存储形式.....33
2.1.4　基本字符集、关键字和标识符.....33
2.2　常量与变量.....34
2.2.1　常量和符号常量.....34
2.2.2　变量.....35
2.2.3　整型变量.....35
2.2.4　实型变量.....36
2.2.5　字符常量与变量.....36
2.2.6　字符串常量.....37
2.3　运算符和表达式.....37
2.3.1　算术运算符和算术表达式.....37
2.3.2　关系运算.....38
2.3.3　逻辑运算.....39
2.3.4　自增、自减运算.....40
2.3.5　赋值运算.....40
2.3.6　条件运算.....41
2.3.7　其他运算.....42
2.3.8　表达式和表达式语句.....43
2.4　类型转换.....47
2.4.1　类型自动转换.....47
2.4.2　赋值转换.....47
2.4.3　强制类型转换.....48
2.5　枚举类型.....48
习题.....49
第 3 章　简单程序设计.....52
3.1　流程结构和语句.....52
3.1.1　三种流程结构.....52
3.1.2　C 的语句概述.....54
3.2　标准输入输出函数.....56
3.2.1　字符输入和输出函数.....56
3.2.2　格式输出函数 printf.....58
3.2.3　格式输入函数 scanf.....63
3.3　简单程序设计举例.....68
习题.....69
第 4 章　流程控制.....72
4.1　复合语句.....72
4.2　if 语句.....75
4.2.1　if 语句实现单分支结构.....75
4.2.2　if～else～语句实现双分支结构.....78
4.2.3　if～else～if～else～…语句实现多分支结构.....81
4.2.4　if 语句的嵌套.....85
4.3　switch 语句.....87
4.4　while 语句.....94
4.5　for 语句.....98

4.6 do-while 语句 103
4.7 多重循环 106
4.8 break 语句、continue 语句和 goto 语句 108
4.8.1 break 语句 108
4.8.2 continue 语句 110
4.8.3 goto 语句 112
4.9 程序举例 113
习题 121
第 5 章 函数与程序结构 124
5.1 C 程序的一般结构 124
5.1.1 模块化程序设计 124
5.1.2 C 程序的一般结构 125
5.2 函数的定义与调用 127
5.2.1 函数的定义 127
5.2.2 函数的调用 130
5.2.3 函数的参数传递 134
5.3 局部变量与全局变量 137
5.3.1 局部变量 137
5.3.2 全局变量 139
5.4 变量的存储类型 141
5.4.1 存储类型区分符 141
5.4.2 自动变量 142
5.4.3 静态变量 144
5.4.4 外部变量 146
5.4.5 寄存器变量 147
5.4.6 存储类型小结 148
5.5 函数的嵌套与递归调用 151
5.5.1 函数的嵌套调用 151
5.5.2 函数的递归调用 152
5.6 内部函数与外部函数 156
5.6.1 内部函数 156
5.6.2 外部函数 157
5.7 编译预处理 157
5.7.1 宏定义 158
5.7.2 文件包含 162
5.7.3 条件编译 164
5.8 程序举例 165
习题 170
第 6 章 数组 172
6.1 数组说明 172
6.1.1 数组的说明 172
6.1.2 数组引用 173
6.1.3 数组的初始化 173
6.1.4 数组的运算（排序与查找） 176
6.2 字符数组 178
6.2.1 字符数组的说明与引用 178
6.2.2 字符数组的初始化 180
6.2.3 字符串和字符串结束标志 181
6.2.4 字符数组的输入输出 182
6.2.5 字符串处理函数 184
6.3 多维数组 189
6.3.1 多维数组的说明、引用和存储结构 189
6.3.2 多维数组的初始化 190
6.3.3 多维数组的运算 191
6.3.4 字符串数组 193
6.4 数组作函数的参数 194
6.4.1 数组元素作函数的参数 194
6.4.2 数组名作函数的参数 194
6.4.3 二维数组作函数的参数 196
6.5 综合实例 197
习题 202
第 7 章 指针 203
7.1 指针的概念 203
7.1.1 变量的地址和指针变量 203
7.1.2 指针说明和指针对象的引用 204
7.2 指针参数 208
7.3 数组的指针表示 210
7.3.1 一维数组的指针表示 211
7.3.2 数组作函数参数时的指针表示 213
7.3.3 字符数组的指针表示 215
7.3.4 多维数组的指针表示与指向数组的指针 217
7.4 指针数组 221
7.4.1 指针数组的概念 221
7.4.2 指针变量的指针 223
7.4.3 main 函数的参数 224
7.5 函数的指针 225

7.6 指针函数 227
7.7 指针相关运算 228
7.8 程序举例 231
习题 235
第 8 章 结构和联合 236
8.1 结构的说明和引用 236
8.1.1 结构说明 236
8.1.2 结构的引用 238
8.2 结构的指针 240
8.3 结构和函数 242
8.4 结构数组 244
8.4.1 结构数组的说明，引用和初始化 244
8.4.2 结构数组作函数参数 250
8.4.3 sizeof 运算符 251
8.4.4 用结构的指针引用结构数组的成员 252
8.5 结构和指针的应用 253
8.6 联合（共用体） 259
8.7 用 typedef 定义类型名 262
习题 263
第 9 章 位运算 266
9.1 位运算的概念 266
9.1.1 字节与位 266
9.1.2 补码 266
9.2 二进制位运算 268
9.2.1 二进制位运算 268
9.2.2 位复合赋值运算符 273
9.3 位段 274
9.3.1 位段的定义与位段中数据的引用 274
9.3.2 对于位段使用的几点说明 276
9.3.3 应用举例 277
9.4 综合应用举例 278
习题 280
第 10 章 输入输出与低层接口 281
10.1 概述 281
10.2 流式文件输入输出 282
10.2.1 文件的打开与关闭 282
10.2.2 文件的读写 284
10.2.3 文件的随机存取 289
10.2.4 其他有关函数 291
10.3 输入输出的低层接口 292
10.3.1 文件的创建、打开、关闭和删除 292
10.3.2 文件的读、写 293
习题 293
附录 A C 关键字 294
附录 B C 运算符的优先级与结合性 295
附录 C 常用字符与 ASCII 码对照表 296
附录 D 常用 ANSI C 标准库函数 297

第 1 章　程序设计技术概述

本章要点

- 算法的定义及其表示方法
- 结构化程序设计方法
- 面向对象的程序设计方法的特点、开发过程
- Visual C++ 6.0 简介

通过本章的学习，使读者对程序设计技术的相关概念和技术有一个全面的了解。

1.1　程序设计技术概要

1.1.1　算法

1. 算法定义与特征

（1）算法定义

人们在科学实验、生产斗争和社会实践中有大量问题需要求解，如科学计算、数据处理及各种管理问题等。要解决这些问题，首先需要分析所研究的对象，提出对问题的形式化定义和给出求解方法的形式描述。对问题的形式化定义叫做数学模型，而对问题求解方法的形式描述称为算法。

也可以说，算法是解题方法的精确描述。算法具有如下性质：解题算法是一有穷动作序列；动作序列仅有一个初始动作；序列中每一动作仅有一个后续动作；序列终止表示问题得到解决或问题没有解答。

算法的非形式定义如下：

一个算法就是一个有穷规则的集合，其中的规则规定了一个解决某一特定类型问题的运算序列。

（2）算法的特征

算法是用来描述解题方法的。任何在计算机上处理的问题都需要用算法进行描述，然后在一个可执行指令的计算机上处理，最后得到结果。作为能解决实际问题的算法，必须具备以下特征：

- 有穷性。一个算法在执行有穷步后一定结束，即一个算法所包含的计算步骤是有限的。
- 确定性。算法的每一个步骤必须经过明确的定义，即算法中所要执行的动作应有严格的规定，而没有歧义性。
- 可行性。算法中要执行的运算和操作是最基本的，它们都能够精确地进行。这样，经过算法描述的一系列步骤的确切动作，最后得到正确的结果。
- 输入。算法有零个或多个输入，即算法开始执行之前，要设置好初始值。

- 输出。算法有一个或多个输出，这一输出就是算法的最终结果。该结果是与输入有关并经过确定的计算步骤后得到的。

2. 算法的表示方法

算法是解题过程的精确描述，那么，用什么描述这一解题过程呢？要描述必须采用某种语言，而语言有多种形式，如自然语言、流程图、伪代码和计算机程序设计语言等。

（1）自然语言

自然语言是人们日常使用的语言，如汉语、英语、日语等，自然语言的规则是人们所使用语言的规则。用自然语言描述算法是最受欢迎的，因为它直观、通俗易懂、为人们所熟悉。当然，用自然语言来描述解题过程，可能会产生歧义性，容易导致算法执行的不确定性。另外，用自然语言表示的算法要翻译成计算机所能认识、理解的形式是不容易的，要做大量的工作。

（2）流程图

流程图是用各种几何图形（如方框、菱形框）、流线及文字说明描述计算过程的框图。流程图是描述算法的常用工具。它的优点是直观，能清楚地表达设计者的思路，清楚易懂。流程图表示的算法不依赖于任何具体的计算机和计算机程序设计语言，有利于程序设计工作。

由于程序的结构主要包括顺序结构、选择结构和循环结构，而流程图又能方便地表示这 3 种结构，所以，用流程图描述的算法过程通过选择合适的程序设计语言后，形成的程序能方便地在计算机上运行。

（3）伪代码

伪代码是界于自然语言和计算机语言之间的、用文字和符号描述算法的一种语言形式，是描述算法的工具。伪代码具有计算机语言的形式，例如有函数、过程的描述，有输入、输出的描述等。伪代码描述的算法侧重于描述算法应完成的主要功能、应满足的条件，而不注重在计算机上实现的细节，有的伪代码中还可以掺杂自然语言的描述。所以，用伪代码写成的算法描述格式紧凑、易于理解，而且便于向计算机程序设计语言描述的算法过渡。

（4）计算机程序设计语言

计算机程序设计语言是编写计算机程序所用的语言。用自然语言、流程图和伪代码描述的算法，计算机是不认识的，这样的算法还必须转换为用计算机程序设计语言描述的程序，计算机才能理解和执行。

程序设计语言按级别可分为高级语言和低级语言。关于程序设计语言的进一步讨论，将在 1.1.3 中进行。

3. 算法分析

为了解决计算机上的各类应用问题，需要用算法描述解题方法。一个算法必须用一个合适的程序设计语言书写成程序，然后才能在计算机上处理，并得到最后的结果。

这里有一个重要的问题是，用算法描述的解题方法是否有效，即需要对算法的效率进行分析。算法对应的程序要到一个具体的计算机上去执行，一个给定的算法在一个具体的计算机上执行的时间是多少？它占用的计算机的存储空间又是多少？另外，对一个给定的

应用问题，如果可用多种算法进行描述，那么，不同的算法各自所需的计算时间和占用的计算机存储空间的大小又各是多少？

算法分析的描述为：算法分析（analysis of algorithms）是对一个算法所需的计算时间和占用的存储空间作定量的分析。

经过算法分析工作，不仅可以预计所设计的算法在什么样的计算机上运行是有效的，还可以分析最好、最坏、平均情况下算法执行的效果。如果为同一问题设计了不同的算法，经过算法分析后可以对这些不同算法的有效性作出比较和判断。

在算法分析中，一般应考虑的 3 个问题是：算法的时间复杂度；算法的空间复杂度；算法是否便于阅读、修改和测试。

要分析一个算法，首先应确定该算法所需的计算时间。算法的时间复杂度指的是：算法有关操作的次数的度量，用 T(n)表示。其中，T 是 Time 的第一个字母，n 是问题规模的大小。

例如，一个 n 次循环程序，每次循环做 a+b->a 的操作，该程序计算的次数为 n；而在累加求和中，n 表示要计算的加数的个数。

在算法的时间复杂度分析中，用 O(f(n))作为 T(n)的上界函数。其中，大 O 作为算法性能描述的工具，表示计算时间的上界函数。f(n)是在分析中确定的某个形式简单的函数，并且是关于正整数 n 的函数，如 logn、2n、n!等，它是独立于机器和语言的函数。

对于 T(n)和 f(n)而言，若存在两个正整数 n0 和 C，当 n->n0 时，有

$|T(n)| \leqslant C|f(n)|$

则算法时间复杂度 T (n)可表示为

$T(n) = O(f(n))$

其含义是：一个算法具有 O(f(n))的计算时间，指的是该算法的计算时间总是小于|f(n)|的常数倍。所以可以说，f(n)是计算时间 T (n)的一个上界函数，T(n)的数量级是 f(n)。

常用的大 O 表示形式有：

- O(1)　　称为常数级
- O(logn)　称为对数级
- O(n)　　称为线性级
- O(n ^ 2)　称为二次多项式级
- O(2 ^ n)　称为指数级
- O(n!)　　称为阶乘级

上述表示的计算时间复杂度从上到下越来越复杂。一个算法的计算时间复杂度应尽可能选用多项式级、线性级及以下的时间复杂度，因为太复杂的算法的实现是非常困难的，且这样的算法也没有实用价值。

算法的空间复杂度是指算法在计算机上的执行过程中所占用的存储空间的大小，用 S(n)表示，S 是英文单词 Space 的第一个字母，n 为问题规模的大小。算法的空间复杂度与时间复杂度的表示方法类似，为

$S(n) = O(g(n))$

但是本书中主要讲述程序设计的基础性知识，注重程序的设计与实现，对于比较简单的算法一般不再进行算法分析。

1.1.2 数据结构

算法是对各类问题解题方法的描述，对于数值计算和非数值计算而言，都需要处理大量的数据。数据不仅具有属性，而且在大多数情况下互相联系，尤其是在非数值处理领域中更是如此。在程序中，数据的特征被抽象为数据类型和值，而信息及其联系被抽象为数据结构。在计算机领域中，数据结构指的是一类定性数学模型，它是计算机算法设计的基础，在计算科学中占有重要的地位。

1. 数据类型

在计算机中，要处理的数据信息表征为量，它是对客观世界实体的一种抽象。例如，计算学生某科学习成绩平均值的程序要处理的数据是学生的考试成绩、学生总数和平均成绩。它抽取了学生学习成绩这方面的特征，而并不关心其他方面。这个抽象的数据可以表示任何学校、任何一个班级、任一科目的学习成绩的平均值，甚至可以用于某一生产车间中工人产值的平均值。

计算机中使用的量可以分为常量和变量。其中，在计算处理中始终保持值不变的量称为常量，而随时间变化的量称为变量。与量对应的数据存储在计算机存储器中，这些量的值是相应存储字中的内容。那么，如何引用这些值呢？这些值有没有一定的范围呢？为了引用这些值，需对数据引进名的概念，用这个名可以访问相应存储器的字。而且，为了刻画数据的可能取值范围，还引进了类型的概念。例如，C 语言提供的基本数据类型为以下 4 种：

char　　字符型
int　　整数型
float　　单精度浮点数型
double　双精度浮点数型

char、int、float、double 等都是类型的名。其中，类型为 int 的量能取任何整数值，类型为 char 的量能取任何字符值等。类型是值的集合，即类型是可能取值所组成的集合。当然，类型也表征了数据的属性。变量应通过类型说明来规定它的属性，如“int i,j;”说明变量 i、j 为整数型。常量数据从书写形式上就可以了解它的值属于什么类型。

类型概念不仅规定了一个量所能取值的范围，同时也规定了对标定类型的量所能进行的运算。例如，对类型为 int 的量只能进行整型运算。所以，概括起来说，量涉及名、值与类型 3 个方面，而类型则涉及名、值集和运算集 3 个方面。

不同的高级语言所提供的数据类型以及类型定义的方式是不相同的。一般数据类型分为基本数据类型（简单数据类型）和构造数据类型两种。构造数据类型是由基本数据类型按某种方式组合而成的，如数组（array）、指针（pointer）、结构（struct）等。

2. 数据结构的组成

数据结构研究的是数据的组织形式，它涉及 3 个方面的内容：抽象数据结构、内部存储结构和数据结构上所施加的运算。

（1）抽象数据结构

抽象数据结构又称数据的逻辑结构，它是客体之间联系的抽象。组成数据结构的数据

元素是基本数据类型或构造数据类型。数据元素又称为结点，抽象数据结构就是各结点之间的逻辑结构。

抽象数据结构分为线性结构和非线性结构。线性结构有串、栈、队、表等，非线性结构有树、图等。

（2）内部存储结构

内部存储结构又称数据的物理结构，它是数据逻辑结构的物理实现方式，是依赖于计算机的，即是数据的逻辑结构在计算机的存储器中的排布方式。它一般有两种方式：顺序存储结构和链式存储结构。

①顺序存储结构：借助数据元素在存储器中的相对位置来表示数据元素的逻辑关系。

②链式存储结构：借助指针来表示数据元素的逻辑关系。一般情况下，在数据元素中增加了一个或多个指针类型的属性，以表示元素之间的连接关系。

（3）数据结构上所施加的运算

在数据结构上所施加的运算有以下几种：建立数据结构；清除数据结构；插入数据元素；删除数据元素；更新数据元素；查找数据元素；按序重新排列；统计数据元素的个数；判定某个数据结构是否为空，或者是否已达到最大允许的容量。

1.1.3　程序和程序设计语言

1. *程序*

要在计算机系统中求解一个实际问题，就必须将问题的数学模型和算法描述用一种计算机能认识的语言来描述，这就是程序设计语言的指令和语句，计算机执行这一系列指令或语句就能完成预期的任务。

程序是按某种顺序排列的，使计算机能执行某种任务（例如解题、检索数据或对一个系统进行控制等）的指令集合。也可以这样定义：程序是计算机系统中计算任务的处理对象和处理规则的描述。

一个程序具有一个单一的、不可分的结构，它规定了某个数据结构上的一个算法。瑞士著名的计算机科学家尼可莱·沃思（Nikiklaus Wirth）在 1976 年曾说道：

程序=算法＋数据结构

程序的基本目标是实现算法和对初始数据进行处理，从而获得所期望的效果。而算法实现过程是由一系列操作所组成的。算法使用一些最基本的操作（如加、减、乘、除、“与”、“或”、“非”、传送、比较），通过对已知条件进行一步一步的加工、变换，最终实现解题目标。这些操作的执行实现了程序应完成的功能，但这只是问题的一个方面。这些操作的执行顺序正确与否也是至关重要的，而操作之间的执行顺序就是控制结构。有 3 种最基本的控制结构：顺序结构、选择结构、循环结构。

顺序结构是 3 种结构中最简单的一种，即语句按照书写的顺序来依次执行；选择结构又称为分支结构，它是根据计算表达式的值来判断应选择执行哪一个流程的分支；循环结构则是在一定条件下反复执行一段语句的流程结构。这 3 种结构就构成了程序局部模块的基本框架。

1966 年，Bohm 和 Jacopini 证明了任何复杂的算法都可以用顺序、选择、循环 3 种结

构组合而成。所以，这 3 种控制结构称为程序的 3 种基本控制结构。如图 1.1 所示为这 3 种基本控制结构的一般形式，其中（c）和（d）是循环结构的两种形式，（c）称为“直到”型循环，（d）称为“当”型循环。“直到”型循环是首先进入循环体，然后判断条件是否成立，若不成立再返回执行循环体，否则退出循环。而“当”型循环是首先判断条件是否成立，若成立则执行循环体，执行后再判断条件是否成立，若成立则继续执行，否则退出循环。

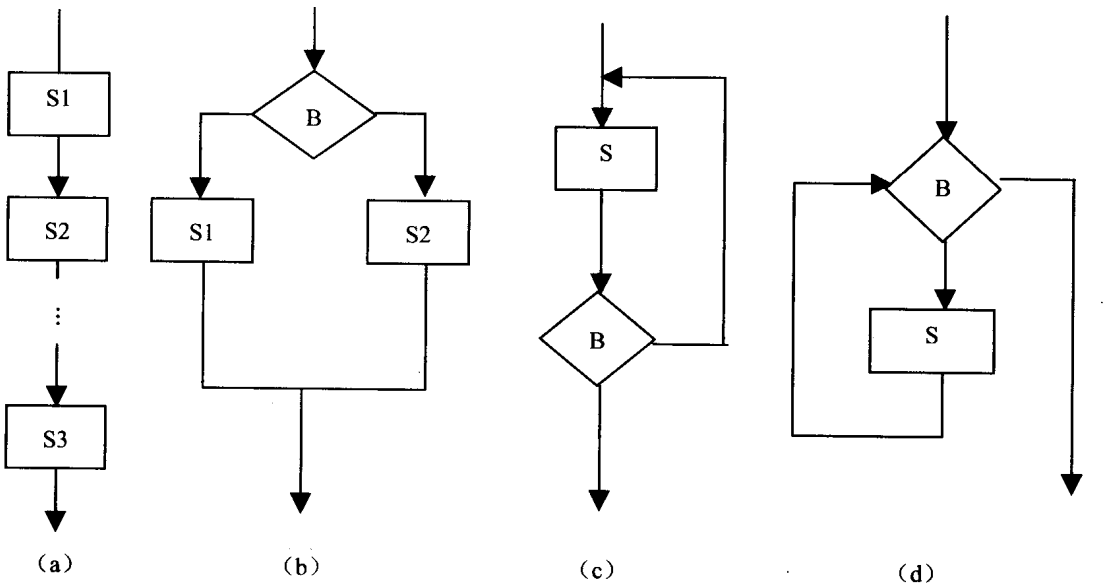

图 1.1　程序的基本控制结构

每个基本控制结构可以看作一个算法单位，整个算法可以由许多个算法单位组合而成。这样的算法容易理解、容易阅读，称为结构化算法。

算法和数据结构是计算机程序的两个基本的、重要的组成部分。算法是程序的核心，而数据结构是对所要加工数据的抽象和组织，将程序要处理的各种松散的数据按某种要求组成数据结构，以便程序能有效地处理。

2. 程序设计语言

程序设计语言是编写计算机程序所用的语言，程序设计语言按语言的级别可分为低级语言和高级语言。低级语言包括机器语言和汇编语言。

（1）机器语言和汇编语言

机器语言是计算机直接使用的语言，由机器指令组成。它使用绝对操作码，即直接使用“0”和“1”两个符号来表示。用机器指令写的程序不需翻译可直接为机器所识别。机器语言效率高，但用机器指令编程序对程序员而言是十分困难的。

汇编语言是一种面向机器的程序设计语言，是一种用符号表示的低级语言，通常是围绕特定的计算机或计算机族而专门设计的。与机器语言相比，汇编语言使用助记码代表机器的操作码，用标识符代替地址码和变址码。汇编语言使程序员摆脱了计算机的一些细节问题，从而可以集中精力考虑程序中的内在联系。

（2）高级语言

高级语言是易为人们所理解的程序设计语言，它是接近日常用语与数学表示方法的程序设计语言。常用的高级语言有 Basic、Pascal、FORTRAN、COBOL、C、Ada 等。

在进行软件开发时，人们往往采用高级语言来编程。有时也部分采用汇编语言编程，这种情况一般在编制系统程序或为了提高效率时才会出现。现在，已基本没有直接用机器语言编程的情况了。

通用的高级语言是在商业、科学、工程计算等方面能广泛应用的程序设计语言。这些语言有大量的软件库，并为大家所熟悉和接受。其中，历史悠久且最基础的通用高级语言有 Basic、FORTRAN、COBOL 等。还有一类通用语言是结构化的程序设计语言，它直接提供结构化的控制结构，具有很强的过程控制能力和数据结构能力，常用的有 Pascal、C 以及 Ada 语言等。

高级语言及语言处理自 20 世纪 50 年代末出现后便发展迅速，现在已有几百种语言存在，按照程序设计语言表达的各种概念和各种结构的不同，可以分为以下几类：

①过程式程序设计语言

又称为面向过程的程序设计语言。过程式程序设计语言具有过程处理的特点，程序主要包含数据结构和算法描述，表现为过程或函数。程序执行被看作各过程调用的序列。Pascal、FORTRAN 和 C 语言就是过程式的程序设计语言。

②面向对象程序设计语言

面向对象程序设计语言是以数据为中心、面向对象的一种语言形式，其程序包含定义的各种对象、类，并规定它们之间传递消息的规律。从程序的执行来看，归结为各对象和它们之间以消息传递方式进行的通信。具有面向对象特征的语言有 C++、Java 等。

③函数式程序设计语言

程序被看作描述输入与输出之间关系的一个数学函数，完全消除状态或变量的概念。Lisp 语言是一种函数式程序设计语言。

（3）语言翻译

计算机只能执行用机器语言编写的程序，对其他语言编写的程序都必须经过翻译，将它们转化为与之等效的机器语言的程序，才能为计算机所认识和理解。根据语言的类别不同，执行方式的不同，常用翻译工具有汇编程序、编译程序和解释程序 3 种。当前用得比较多的、采用面向对象技术、面向网络编程的 Java 语言有新的工作机制，下面将进行简单介绍。

汇编程序是一种翻译程序，其功能是将用汇编语言编写的程序翻译成机器语言程序。汇编程序的特点是，其指令与翻译后的机器语言指令具有一一对应的关系。

编译程序也是一种翻译程序。它将用面向过程语言编写的源程序作为数据接收，经过翻译转换，产生并输出与源程序等效的目标程序。该目标程序可以是直接转换生成的机器代码，也可以是汇编语言描述的程序，若为后者，还需要经过汇编阶段。

解释程序将用高级语言编写的程序作为输入，但并不产生目标程序，而是边解释边执行，这样的翻译程序称为解释程序。在解释程序控制下，按源程序中各语句动态执行的顺序，一次读一个语句，根据该语句的功能，翻译成与该语句对应的机器代码，并立即进行处理。翻译执行完一个语句后，按执行顺序读入下一个语句，直至将全部源程序动态处理