

21世纪高校
计算机系列教程

C语言程序设计



洪维恩 编著

引进台湾原版成熟教材■

注重内容的实用性，培养学生的专业能力■

适合高校电子信息类各专业选用■

经过全国高等院校计算机基础教育研究会著名专家学者、教授的评估与审定■

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



C 语言程序设计

洪维恩 编著

中国铁道出版社

2003年·北京

(京)新登字 063 号

北京市版权局著作合同登记号: 01-2003-5908 号

内 容 简 介

本书从 C 语言的特色和优势出发,介绍 C 语言程序设计的规划、编写、编译执行以及大型程序的开发理念,囊括了 C 语言的数据类型、格式化输入/输出、运算法则、循环和选择语句的运用,函数、数组、字符串、指针、结构体及其他数据类型的应用,文件的处理、位运算处理、大型程序的开发等内容。全书实例丰富,语言简练,展现了 C 语言的精髓,且使用免费的 Dev C++ 环境编写、编译和运行程序,让读者可以摆脱 DOS 环境的 Turbo C,更轻松地学习 C 语言。

本书适合高校计算机及相关专业作为 C 语言程序设计的教材,同时也可作为培训和自学用书。

图书在版编目 (CIP) 数据

C 语言程序设计/洪维恩编者. —北京:中国铁道出版社,2003.10

(21 世纪高校计算机系列教程)

ISBN 7-113-05331-9

I. C… II. 洪… 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(2003)第 092755 号

书 名: C 语言程序设计

作 者: 洪维恩

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 魏 春 张雁芳

责任编辑: 苏 茜 王占清 夏华香

封面设计: 孙天昭

印 刷: 北京市彩桥印刷厂

开 本: 787×1092 1/16 印张: 23.25 字数: 560 千

版 本: 2003 年 11 月第 1 版 2003 年 11 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-05331-9/TP·967

定 价: 28.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

21 世纪高校计算机系列教材 审定委员会

主任：高 林

副主任：丁桂芝 李 畅

委员：（以下排名不分先后）

安淑芝 鲍有文 陈文博 樊月华 陈维兴

顾星海 李秀芬

项目策划：严晓舟 宋旭明

编 辑：魏 春 秦绪好 张雁芳

丛 书 序

21 世纪的今天，以计算机和因特网为代表的信息技术在高速发展，并在社会各行各业、各个层面得到广泛应用，信息技术深刻地改变着人类的生活、工作和思维方式。时代要求每一个人都应当学习信息技术、应用信息技术，因此计算机教育已经发展成为信息技术教育。

随着我国社会主义市场经济和信息技术的高速发展，我国的高校计算机教育已从重视理论教育，走向理论与实践相结合、注重运用知识解决实际问题能力的发展阶段。目前，大学计算机应用技术和高职高专教育还处于探索阶段，与之相适应的教材建设已成为教育的重点。

我国台湾地区的高校计算机应用技术教育发展已相当成熟，在过去的二十几年中，为台湾地区经济发展培养了大批应用型人才，在职业化教育方面积累了丰富的经验，在处理理论与实践的关系以及培养实际应用与操作的技术应用型人才方面都处于领先地位，也符合国际职业教育发展的主流。由于有着相同的历史文化背景，现阶段台湾的高校计算机应用技术教育教材对我们更具有借鉴作用。

因此根据教育部关于引进外文原版教材、推动我国教材改革的精神，通过多方面调查，在充分听取专家意见的基础上，中国铁道出版社以“引进，吸收，创新”为指导思想，力争走出一条新的高校计算机应用技术教育教材发展之路。作为这一决策的第一步，我们经过精心的组织策划，推出了“21 世纪高校计算机系列教程”。

我们本次引进的教材都是现阶段我国台湾地区的科技大学和技职学院正在使用的教材，这些学校是当前台湾地区发展高校计算机应用技术教育的主体学校，我们认为这些教材在编写过程中均采用了理论与实践相结合的方法，注重在学习理论知识的基础上重点培养学生的实践能力，通过一系列实际训练，提高学生各种必备的技术应用能力，使他们一出校门便可发挥其所学专长，成为该领域的专业人才。

全国高等院校计算机基础教育研究会高林教授充分肯定了铁道出版社引进台湾版应用技术教材的思路，并在教材引进和编审过程中给予了全面指导，组织多位专家教授对这套教材进行了审定。专家们一致认为本套教材内容新颖，易教易学，层次配套，符合高校计算机应用技术教育的特点，有利于推动我国高等职业教育的发展，建议在全国范围内推广使用。

本套教材可以作为大学计算机应用技术课程教材以及高职高专、成人高校和面向社会的培训班的教材，也可作为学习计算机应用技术的自学教材。

中国铁道出版社
2003 年 9 月

前 言

C 语言之所以有这么大的魅力，是因为 C 语言的可移植性高，在绝大部分的计算机与操作系统上都可以使用，同时它易学好记，效率很高且语法清晰。C 语言融合了计算机语言里流程控制的特点，使得程序员可以很容易地设计出具有结构化及模块化的应用软件。也因为如此，许多软件与操作系统（如 Unix、MS-Windows 等）均由 C 语言编写。此外，许多高级语言的编译器或解释器也是 C 语言的杰作。当您手握鼠标，挥洒于 Windows 的天地时，想想这个精巧人机接口的背后，可是数十万行 C 语言程序的结晶！本书希望能通过“边做边学”的方式，来使读者学习并熟悉 C 语言。如果稍稍浏览本书的内容，您可以发现文字浅显、易读易懂是本书的一大特色。

本书由台湾博硕文化股份有限公司提供版权，经中国铁道出版社计算机图书项目中心审选，北方交通大学的须德教授、刘娜、陈兰芳、程瑞芬、崔仙翠等同志完成了本书的整稿及编排工作。

本书中的所有范例程序代码、Dev C++编译器及 C 语言在 Visual C++和 Turbo C 中的使用说明可以到天勤网站 <http://www.tqbooks.net> “下载专区”下载。

为方便教学，我们将为选用本书作教材的任课老师免费提供本书的“电子教案素材库”光盘一张。

联系电话：83529867

2003 年 11 月

目 录

第 1 章 认识 C 语言	1
1-1 缘起	2
1-2 C 语言的特点	2
1-3 C 语言与您	4
1-4 程序的规划与操作	5
1-5 编写第一个 C 语言程序	9
1-6 程序代码的编译、执行与保存	9
1-7 编译与执行的过程	10
1-8 本书的编排与惯例	11
课后习题	12
第 2 章 简单的 C 语言	13
2-1 一个简单的例子	14
2-2 C 语言解析	15
2-2-1 #include 宏命令及头文件	15
2-2-2 main() 函数	17
2-2-3 花括号、段及主体	17
2-2-4 变量	17
2-2-5 变量的赋值	18
2-2-6 为什么要声明变量	19
2-2-7 printf() 函数	19
2-3 标识符及关键字	20
2-3-1 标识符	20
2-3-2 关键字	20
2-4 再举一个简单的例子	20
2-5 调试	21
2-5-1 语法错误 (Syntax Error)	21
2-5-2 语义错误 (Semantic Error)	21
2-6 提高程序的可读性	22
课后习题	23
第 3 章 变量与数据类型	25
3-1 简单的实例	26
3-2 变量与常量	26
3-3 基本的数据类型	26
3-3-1 整型类型 int	27
3-3-2 字符类型 char	29
3-3-3 浮点数类型 float	32



3-3-4 双精度浮点数类型 double	33
3-4 数据类型的转换	33
课后习题	35
第 4 章 格式化的输入与输出	36
4-1 printf() 函数	37
4-1-1 使用 printf() 函数	37
4-1-2 printf() 函数的修饰符 (Modifier)	39
4-1-3 不同类型的输出	41
4-1-4 printf() 函数的换行	43
4-2 scanf() 函数	44
4-2-1 scanf() 函数的使用	45
4-2-2 scanf() 函数的使用方式	47
4-2-3 字符串的输入	50
4-3 输入/输出字符的函数	52
4-3-1 getchar() 与 putchar() 函数	52
4-3-2 getche() 与 putch() 函数	53
4-3-3 getch() 函数	53
课后习题	54
第 5 章 运算符、表达式与语句	56
5-1 运算符	57
5-1-1 赋值运算符	57
5-1-2 一元运算符	58
5-1-3 算术运算符	59
5-1-4 关系运算符与 if 语句	61
5-1-5 自增与自减运算符	62
5-1-6 逻辑运算符	63
5-1-7 括号运算符	65
5-2 运算符的优先级	65
5-3 表达式	66
5-4 表达式的类型转换	69
5-4-1 类型转换的规则	69
5-4-2 强制类型转换	70
课后习题	71
第 6 章 循环	73
6-1 结构化程序设计	74
6-1-1 顺序结构	74
6-1-2 选择结构	75
6-1-3 循环结构	76
6-2 for 循环	76
6-3 while 循环	79

6-3-1 使用 while 循环	80
6-3-2 无穷循环的产生	82
6-4 do while 循环	83
6-5 空循环	86
6-6 我要使用哪一种循环	87
6-7 嵌套循环	88
6-8 循环的跳离	93
6-8-1 break 语句	94
6-8-2 continue 语句	95
课后习题	97
第 7 章 选择性语句	99
7-1 我的程序会转弯——if 语句	100
7-2 其他选择	102
7-2-1 if-else 语句	102
7-2-2 更多的选择——嵌套 if 语句	105
7-2-3 else if 语句	106
7-2-4 非常选择——if 与 else 的配对问题	108
7-3 if 的简洁版——条件运算符	110
7-4 更好用的多重选择——switch 语句	112
7-4-1 switch 语句与 break 语句	112
7-4-2 不加 break 语句的 switch 语句	118
7-5 goto 语句	119
课后习题	121
第 8 章 函数与预处理命令	122
8-1 简单的函数	123
8-2 函数的基本结构	124
8-2-1 函数原型的声明、编写与调用	124
8-2-2 函数的自变量与参数	126
8-2-3 函数的常量返回值	128
8-3 变量的等级	132
8-3-1 局部变量	132
8-3-3 外部变量	135
8-3-4 静态外部变量	137
8-3-5 寄存器变量	139
8-4 同时使用多个函数	141
8-4-1 调用多个函数	141
8-4-2 函数之间的相互调用	142
8-4-3 递归函数	143
8-5 预处理命令——#define	148
8-5-1 #define 预处理	148
8-5-2 为什么要用#define	149
8-5-3 const 修饰符	150

8-5-4	#define 的另一功能——宏——取代简单的函数.....	150
8-5-5	使用自变量的宏.....	151
8-5-6	宏号的使用.....	152
8-5-7	使用函数还是使用宏.....	153
8-6	再来看看#include 预处理命令.....	153
8-6-1	使用自定义的头文件.....	153
8-6-2	标准的头文件.....	155
8-6-3	头文件与函数原型.....	155
	课后习题.....	155
第 9 章	数组与字符串.....	157
9-1	一维数组.....	158
9-1-1	数组的声明.....	158
9-1-2	数组中元素的表示方法.....	159
9-1-3	数组初始化赋值.....	159
9-1-4	数组的输入与输出.....	160
9-1-5	数组越界的检查.....	163
9-2	二维数组以上的多维数组.....	165
9-2-1	二维数组的声明与初始化赋值.....	165
9-2-2	二维数组元素的引用及存取.....	166
9-2-3	多维数组.....	168
9-3	传递数组给函数.....	169
9-3-1	以一维数组为自变量来传递.....	169
9-3-2	冒泡排序法.....	174
9-3-3	传递多维数组.....	178
9-3-4	传递“值”还是“地址”到函数.....	181
9-4	字符串.....	184
9-4-1	字符串常数.....	184
9-4-2	字符串的声明与初始化的赋值.....	184
9-5	字符串的输入与输出函数.....	185
9-5-1	scanf()与printf()函数.....	185
9-5-2	gets()与puts()函数.....	186
9-6	字符串数组.....	187
9-6-1	字符串数组的声明与初始化赋值.....	187
9-6-2	字符串数组元素的引用及存取.....	188
	课后习题.....	192
第 10 章	指针.....	195
10-1	指针概述.....	196
10-1-1	什么是指针.....	196
10-1-2	为什么要用指针.....	198
10-2	指针变量.....	198
10-2-1	指针变量的定义.....	198
10-2-2	指针变量的使用.....	199

10-3	指针运算符	202
10-3-1	地址运算符&	202
10-3-2	按照地址取值运算符*	203
10-3-3	定义指针变量所指类型的重要性	203
10-4	指针的运算	204
10-4-1	指针的赋值运算与赋值	204
10-4-2	指针的加法与减法运算	206
10-4-3	指针的减法运算	208
10-5	指针与函数	208
10-6	指针与数组	215
10-6-1	指针与数组的关系	216
10-6-2	字符串数组与指针数组	219
10-7	指向指针的指针——双重指针	221
	课后习题	226
第 11 章 结构体与其他数据类型		227
11-1	结构体	228
11-1-1	结构体的声明	228
11-1-2	结构体变量的使用及初始化赋值	229
11-2	嵌套结构体	232
11-3	结构体数组	234
11-4	结构体指针	238
11-5	以结构体为自变量传递到函数	240
11-5-1	将整个结构体传递到函数	241
11-5-2	将结构体字段分别传递	242
11-5-3	传递结构体的地址	243
11-6	共用体 (Union)	245
11-6-1	共用体的定义及声明	246
11-6-2	共用体与结构体的差异	247
11-6-3	共用体的使用及初始化赋值	249
11-7	枚举类型 (Enumeration)	251
11-7-1	枚举类型的定义及声明	252
11-7-2	枚举类型的使用及初始化赋值	252
11-8	使用自定义的类型——typedef	258
	课后习题	261
第 12 章 文件		263
12-1	文件的概念	264
12-2	文件的操作方式	264
12-3	有缓冲区的文件处理函数	265
12-3-1	文件处理函数的整理	267
12-3-2	文件处理函数的练习	268
12-4	无缓冲区的文件处理函数	278
12-4-1	文件处理函数的整理	279



12-4-2 文件处理函数的练习.....	280
12-5 二进制文件的使用	283
12-5-1 二进制文件有缓冲区函数使用.....	283
12-5-2 二进制文件无缓冲区函数使用.....	287
12-6 文本模式及二进制模式的比较	290
12-6-1 以文本模式存储数值.....	290
12-6-2 以二进制模式存储数值.....	292
12-6-3 换行与文件结束的讨论.....	294
12-6-4 输出相对应字符的十六进制码.....	296
12-7 顺序存取与随机存取	298
12-7-1 顺序存取 (Sequential Access)	298
12-7-2 随机存取 (Random Access)	299
课后习题.....	302
第 13 章 大型程序的发展	303
13-1 程序的模块化	304
13-2 各自编译的实例	304
13-3 条件式编译	307
13-3-1 #if、#else 与#endif 命令.....	308
13-3-2 #elif 命令.....	308
13-3-3 #ifdef 和#ifndef 命令.....	309
13-3-4 #undef 命令.....	310
13-4 命令行参数的使用	311
课后习题.....	314
第 14 章 位运算.....	315
14-1 数字系统、位和字节.....	316
14-2 二进制系统	317
14-2-1 二进制的整数表示方式.....	317
14-2-2 带有正负号的二进制整数.....	317
14-3 其他的进制系统	320
14-3-1 八进制系统.....	320
14-3-2 十六进制系统.....	321
14-4 位运算符	322
14-4-1 位逻辑运算符.....	322
14-4-2 位位移运算符.....	325
14-5 位段	329
课后习题.....	331
附录.....	333
附录 A Dev C++的使用.....	334
附录 B 常用的函数库.....	347
附录 C ASCII 码表.....	356

1

认识 C 语言

C 语言尤如经过瘦身与塑身后的一种计算机程序语言一样，它娇小、匀称，功能强而且效率高，因而深受程序员及信息界人士的喜爱。目前有许多应用程序诸如图像处理程序、数据结构、人机接口控制及数值分析等都由 C 语言来编写，由此可见 C 语言受欢迎的程度。本章将带您初探 C 语言的世界，并教您编写第一个 C 语言程序。

- 1-1 缘起
- 1-2 C 语言的特点
- 1-3 C 语言与您
- 1-4 程序的规划与操作
- 1-5 编写第一个 C 语言程序
- 1-6 程序代码的编译、执行与保存
- 1-7 编译与执行的过程
- 1-8 本书的编排与惯例



1-1 缘起

美国的贝尔实验室（Bell Laboratory）成立迄今，成果极其丰硕而且造就了不少人才，C 语言即是在这个实验室里由 Dennis Ritchie 于 1972 年开发出来的。C 语言的前身为 B 语言，最早是用来编写 DEC PDP-11 计算机的系统程序。这个系统程序与人们所熟悉的 Unix 操作系统有着密不可分的关系。原本 C 语言只能在大型计算机里执行，现在已成功地移植到个人计算机里，而且有不同的版本出现，其中人们比较熟悉的有 Turbo C、Microsoft C、Quick C 与 Lattice C 等。

1-2 C 语言的特点

任何一种计算机语言的发展均有其目的。例如 BASIC 语言，其主要的目的是要让计算机的初学者也可以很容易地编写程序，所以其语法近似英文，而且浅显易懂；此外，应科学计算与商业用途的需要，FORTRAN 与 COBOL 语言也应运而生。其他的高级语言如 PASCAL 等也有其特定的用途。但这些语言常因发展背景与语言本身的限制而无法兼顾实用与性能。C 语言的诞生恰恰可以弥补上述的缺憾，因而现在已成了程序员的最佳工具。一般而言，C 具有下列几项特点：

1. 高性能的编译式语言

一般来说，当源程序代码编写完成后，必须转换成机器所能理解的语言，才能正确地执行。所有的程序语言中，都附有这种转换程序，而转换程序可以大致分为两种，即解释器（Interpreter）与编译器（Compiler）。

所谓的解释器，就是当我们要执行程序时，这个解释器就会逐行的检查程序的语法，如果没有错误，再直接执行该程序（如果碰到错误就会立刻中断执行），直到程序完毕（见图 1-1）。利用这种方式完成的程序语言，最著名的就属 BASIC。由于解释器只需要将程序逐行翻译并访问源程序即可，所以所占用的内存较少，但是每一行程序在执行前才被翻译，将导致翻译时间会延迟执行时间，因此执行的速度会变慢，效率也较低。

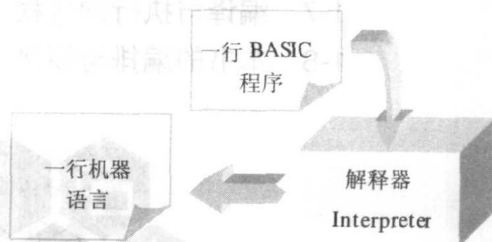


图 1-1 解释器会逐行检查程序的语法，再直接执行该程序，直到程序完毕

然而编译器则会将整个程序都检查完成，先产生一个目标文件（OBJ 文件），将其他要连接进来的程序链接后，再执行该程序（见图 1-2），源程序每修改一次，就必须重新编译，才能保持其可执行文件为最新的状况，同时，在执行的过程中也不需要因为等待程序的编译而中断。经过编译器所编译出来的程序，在执行时不需要再翻译，因此，执行效率与速度远高于解释程序，但是，由于编译器会产生诸如目标文件等的相关文件，也较占用内存空间。常见的编译式程序语言有 C、COBOL、PASCAL 等。然而 C 的执行效率与使用的普遍性，远

远超过其他的程序语言。

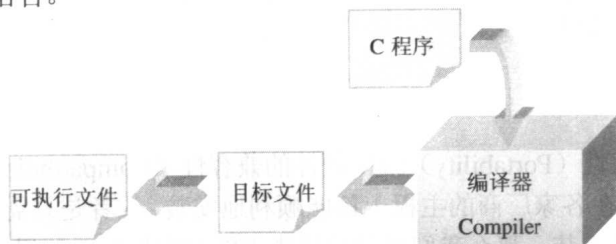


图 1-2 编译器先产生目标文件，再执行该程序

2. 介于高级语言与低级语言之间的一种语言

程序语言按其特点可大致分为两类——低级语言与高级语言。

低级语言在计算机里的执行效率相当高，而且对于硬件（如鼠标、键盘等）控制的程度相当好，但对用户而言，它却生涩难懂，编写、阅读与维护都不容易。

高级语言为叙述性语言，它与人类所惯用的语法比较接近，所以容易编写、排错，但是相对的，它对硬件的控制能力却比较差，执行效率也远不及低级语言。常见的高级语言有 BASIC、FORTRAN、PASCAL 以及 COBOL 等。

C 语言不但具有低级语言的优点（对硬件的控制能力强），同时也兼顾了高级语言的特点（易于排错、编写），所以有人称之为“中级语言”。此外，C 语言还可以很容易地与汇编语言链接，利用低级语言的特点来提高程序代码的执行效率。

3. 灵活的控制流程与结构化的格式

C 语言是性能高、语法清晰的语言。它融合了计算机语言里流程控制的特点，使得程序员可以很容易地设计出具有结构化及模块化的程序语言，如图 1-3 所示。

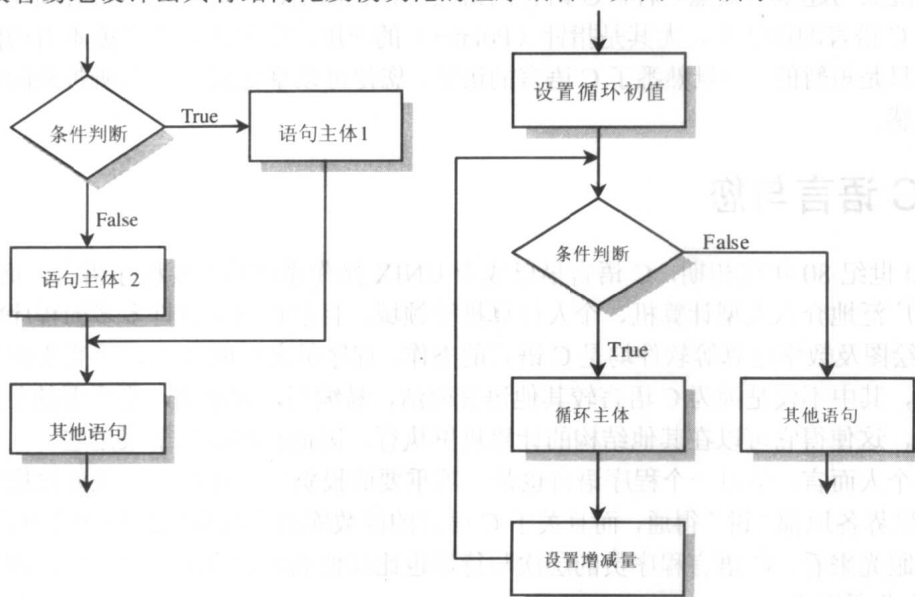


图 1-3 灵活的控制流程与结构化的格式

由于 C 语言的高性能与灵活性，许多操作系统（如 UNIX、MS-Windows 等）均由它编写。此外，许多高级语言的编译器（Compiler）或解释器（Interpreter）也是 C 语言的杰作。



所以当您手握鼠标，挥洒于 Windows 的天地时，是否会意识到这个精巧的人机接口的背后却是数十万行 C 语言的结晶！

4. 可移植性佳

程序语言的可移植性 (Portability) 就像硬件的兼容性 (Compatibility) 一样。举例来说，如果您买了一块声卡，在每家厂商的主板上都能顺利地安装，或者是只需要调整一下设置 (如 Jumper、IRQ 等) 即可安装，那么我们就说这块声卡的兼容性高。如果仅可在特定的主板上使用，那么这块声卡的兼容性就差。

同样地，程序语言的可移植性好，意味着在某一系统所编写的语言可以在少量修改或完全不修改的情况下即可在另一个操作系统里执行。C 语言可以说是一个可移植性极好的语言，当您想跨越平台来执行 C 语言时 (如将 UNIX 里的 C 程序代码移植到 Windows 的环境里执行) 通常只要修改极少部分的程序代码，再重新编译即可执行。此外，提供 C 编译器的系统近 50 种，从早期的 Apple II 到超级计算机 Cray，均可找到 C 编译器的踪迹。

5. 为程序员所设计的语言

C 语言可以说是专为程序员所设计的语言。它可以让您直接按照内存的地址来存取变量，以提高程序执行的效率，此外，C 语言也提供了丰富的运算符 (Operator)，使得 C 语言的语法更为简洁、有力。更方便的是，在大多数的 C 语言环境里都提供了已编写好的函数库 (Library)，包含了许多 C 语言函数，以供程序员使用而无需重新编写程序代码。

6. C 语言的另一面

通常，一个优点的背后往往隐含着它的缺点，就如同一台功能超强的笔记本电脑，往往有着体积过大与超重的缺点一样。C 语言的语法严谨、简洁，相对的用户就必须花更多的心思在学习 C 语言的语法上，尤其是指针 (Pointer) 的应用，常常让初学者摸不着边际。但我们相信这只是短暂的，一旦熟悉了 C 语言的语法，您便可以享受到 C 语言所带来的便利性与超速的快感。

1-3 C 语言与您

在 20 世纪 80 年代初期，C 语言早已成为 UNIX 操作系统的主要程序语言，而现在，C 语言更是广泛地介入大型计算机、个人计算机等领域。目前市面上许许多多的程序游戏、文字处理、绘图及数学运算等软件均是 C 语言的杰作。程序员之所以选择 C 语言来作为软件的开发工具，其中不仅是因为 C 语言较其他语言简洁，易编写，效率高，更重要的是，C 语言易于修改，这使得它可以在其他结构的计算机里执行，因而程序员乐于采用它。

对于个人而言，学习一个程序语言也是一项重要的投资。学习 C 语言就好比您学英文一样，走到世界各地都“讲”得通，而且关于 C 语言的函数库的获取也较其他计算机语言容易。以现实的眼光来看，C 语言程序员的层次与待遇也比其他高级语言的程序员高，而且比较容易被 IT 企业所聘用。

再者，明日之星 C++ 或者是 Java 等语言均是以 C 为基础，再加上面向对象的功能，使得 C++ 与 Java 更加活跃于窗口与网络的程序设计 (如 Visual C++、C++ Builder 及 Microsoft Visual J++ 等)，如图 1-4 所示。因此，C 语言的投资回报相当高，值得您花时间去研究。当然，也

别让您在您的履历表里缺席。要学好 C 语言，就从现在开始吧！

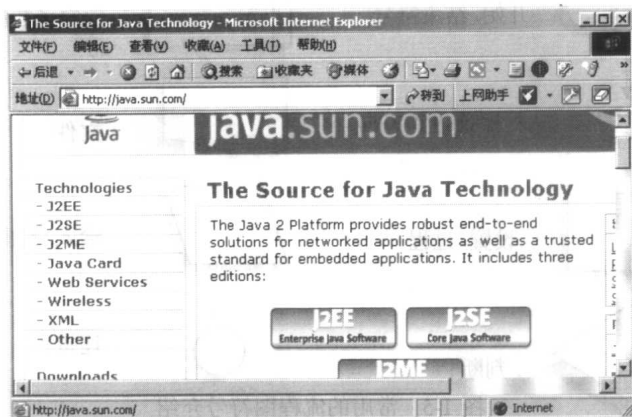


图 1-4 Java 语言可以用来做 Web 画面的程序设计，它的语言源自于 C++，语法与 C 极为相似

1-4 程序的规划与操作

一般来说，程序的设计分为自顶向下法（Top-down Approach）与自底向上法（Bottom-up Approach）。在程序设计的过程中，如果能够将问题分解成多个模块（Modules），再将这些模块分别分解成更小的模块，以此类推，直到分解成最容易编写的最小模块为止，这种程序设计的方式即称为自顶向下法（Top-down Approach）。利用“自顶向下”的方式所编写的程序，其结构有层次，容易看懂和维护，同时可以降低开发的成本，但是在程序分解成模块的过程中，可能因此占用较多的内存空间，造成执行时间过长。

如果在程序设计时，先将整个问题里最简单的部分编写出来，再一一结合各个部分以完成整个程序，这种设计的方式就称为自底向上法（Bottom-up Approach）。利用“自底向上”的方式所编写的程序，不太容易看懂和维护，造成程序设计者的负担，反而容易增加开发的成本。

因此在编写程序前的规划就显得相当重要，如果程序的内容很简单，当然可以马上把程序写出来；但是当程序愈大或是愈复杂时，规划的工作就很重要，它可以让程序设计有明确的方向，尤其是被自己程序的逻辑搞得一团糟的时候，有了事前的规划流程，就可以根据这个流程来一步一步设计出理想的程序。

除了这个好处之外，还可以养成规划程序的习惯，可以发现程序会简洁许多。这也意味着程序执行的速度将会更快、更有效率。了解程序规划的重要性后，我们再来看看设计程序所需的 6 大步骤：

1. 规划程序

首先，您必须明确编写这个程序的目的、程序的用户对象以及需求度，如计算员工每个月的工资、绘制图表、数据排序等，再根据这些数据及程序语言的特性，选择一个合适的程序语言，来达到您设计程序的目的。您可以在纸上先绘制出简单的流程图，将程序的起始到结束的过程写出，一方面，便于理清程序的思路；另一方面，可以根据这个流程图进行编写程序的工作。图 1-5 是绘制流程图时常会用到的流程图符号介绍。

我们以一个日常生活的例子“出门时如果下雨就带伞，否则戴太阳眼镜”，简单地说明如何绘制程序流程图。