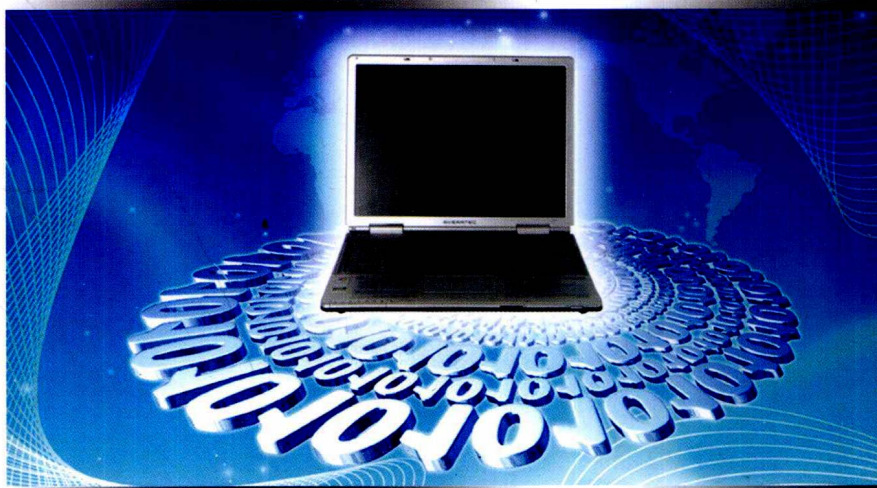
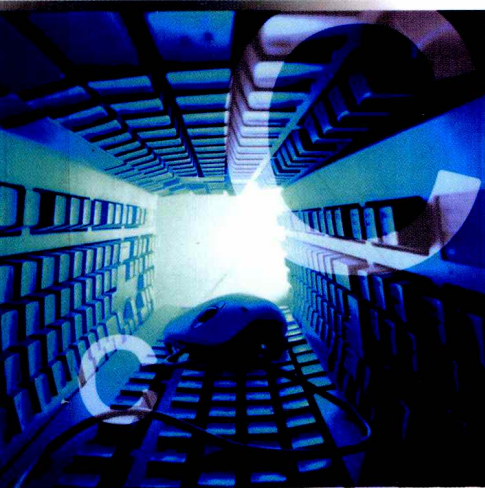




普通高等教育“十一五”规划教材

C 语言程序设计

C YUYANCHENGXUSHEJI



主 编 马学文 谢华成
副主编 王新霞 刘 琦 李国梁



中国水利水电出版社
www.waterpub.com.cn

普通高等教育“十一五”规划教材

C 语言程序设计

主 编 马学文 谢华成

副主编 王新霞 刘 琦 李国梁



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书以现代C语言为主导,以流行的Visual C++ 6.0为编译环境,全面介绍了C语言程序设计及经典算法。全书共分10章,内容包括程序设计基础,数据类型,运算符与表达式,顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组,函数,指针,结构体与共同体和文件处理等。书中所有实例均在Visual C++ 6.0编译环境下调试通过。

本书结构严谨,文笔流畅,例题丰富,分析透彻,文档规范,将知识传授与能力训练相结合,重点培养学生分析问题和解决问题的能力,可以作为高等院校非计算机专业C语言程序设计教材,也可作为计算机应用能力资格考试的辅导材料和程序设计人员的参考书。

本书配有电子教案,读者可以从中国水利水电出版社网站和万水书苑免费下载,网址为:
<http://www.waterpub.com.cn/softdown/>和 <http://www.wsbookshow.com>。

C语言程序设计 / 马学文, 谢华成主编. -- 北京 :
中国水利水电出版社, 2010.8
普通高等教育“十一五”规划教材
ISBN 978-7-5084-7755-8

I. ①C… II. ①马… ②谢… III. ①
C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第149923号

策划编辑: 雷顺加/向辉 责任编辑: 李 炎 加工编辑: 樊昭然 封面设计: 李 佳

书 名	普通高等教育“十一五”规划教材 C 语言程序设计
作 者	主 编 马学文 谢华成 副主编 王新霞 刘 琦 李国梁
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	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	北京市天竺颖华印刷厂
规 格	184mm×260mm 16开本 16.5印张 422千字
版 次	2010年8月第1版 2010年8月第1次印刷
印 数	0001—4000册
定 价	28.00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

前 言

信息时代需要信息技术，信息技术已与其他学科相互交融，成为推动社会发展的动力。能够接受高等教育的各专业学生，都必须具备计算机的基础知识和应用能力。计算机的应用能力已经成为高素质人才的必备要素。

近年来，从事软件开发和计算机教育的人员发现 C 语言具有应用面广，使用灵活方便，表达力强和执行效率高等特点，非常适合程序设计初学者作为入门基础。熟练掌握 C 语言的设计人员可以快速掌握其他高级语言，因此 C 语言被计算机专业和非专业应用人员广泛学习和使用。目前，全国许多高校将“C 语言程序设计”作为一门公修计算机基础课程，同时，全国计算机等级考试以及各地区组织的计算机等级考试都把 C 语言列入考试范围。

本书适用于普通本专科高校计算机程序设计基础教材，可供计算机专业和非专业的 C 程序设计基础课教学使用，也适用于程序设计的初学者和想更深入了解 C 语言的人员使用。在本书编写过程中，编者将知识讲授和能力培养有机结合，在讲透彻基础知识的基础上，选取了合适的例程，以便读者加深对基本知识的掌握和理解。内容讲述中，注重培养读者分析问题和程序设计的能力，引导读者形成程序思维模式和良好的编码风格。与同类书相比，本书体现了以下特色：

(1) 入门门槛低，不要求读者掌握其他程序设计语言，本教材中的程序设计思想即程序设计基础。

(2) 内容全面，编排合理。在内容编排上，注意分散难点，便于读者循序渐进地学习。

(3) 详略得当，重点突出。本书主要讲解 C 语言最基本、最常用的内容。

(4) 强化实践，重视应用。本书的讲解以 C 语言的语法、语义为基础，旨在帮助读者掌握 C 语言程序设计的技巧，具备使用程序设计解决实际问题的能力。所以本书结合大学生计算机等级考试，在各章中提供了较多的实例，使读者能够深入理解和有效训练。

(5) 本书以 Visual C++ 6.0 编译环境为实验平台，与 Turbo C 环境相比，该平台方便操作、查错和调试，更重要的是该平台具有工程化的管理模式，对读者的成长更为有益。

本书共分 10 章。第 1 章介绍程序设计基础，第 2 章介绍 C 语言的数据类型与表达式，第 3 章介绍顺序结构程序设计和编译预处理，第 4 章介绍选择结构程序设计，第 5 章介绍循环结构程序设计，第 6 章介绍数组与字符数据处理，第 7 章介绍函数与程序结构，第 8 章介绍指针，第 9 章介绍结构体和共同体，第 10 章介绍文件处理。

本书由马学文、谢华成任主编，王新霞、刘琦、李国梁任副主编。其中马学文编写了前言，第 1 章和第 6 章；谢华成编写了第 8 章和第 10 章；王新霞编写了第 2 章和第 3 章；刘琦编写了第 7 章和第 9 章，李国梁编写了第 4 章和第 5 章，袁晓亚编写了附录部分，参与本书

编写的还有谢蕾、周嫒、谈静、秦乐阳、徐现伟、苏辉、刘坤、陈功平等。

中国水利水电出版社的工作人员为本书的成功出版付出了艰辛的劳动，编者在此对为本书成功出版做出贡献的所有人员表示衷心的感谢。

由于时间仓促，加上作者水平有限，书中难免存在不足之处，欢迎读者批评指正。

作 者

2010年7月

目 录

前言

第1章 程序设计基础	1	2.4.1 整型变量	23
1.1 程序设计语言的发展	1	2.4.2 实型变量	26
1.2 算法与程序设计	1	2.4.3 字符变量	27
1.2.1 算法及其表示	1	2.5 数据间的混合运算	28
1.2.2 结构化程序设计 with 面向对象程序设计	5	2.5.1 各数据类型转换规则	28
1.3 C语言简介	6	2.5.2 自动类型转换与强制转换	29
1.3.1 C语言的产生与发展	6	2.6 变量赋初值	30
1.3.2 C语言的特点与应用领域	6	2.7 运算符和表达式	31
1.4 C语言的结构与执行	6	2.7.1 C语言运算符简介与表达式的类型	31
1.4.1 C语言源程序的结构特点	6	2.7.2 表达式的求值规则	31
1.4.2 高级语言程序的执行过程	7	2.7.3 算术运算符和算术表达式	31
1.4.3 程序设计的开发期与开发习惯	7	2.7.4 赋值运算符和赋值表达式	33
1.5 C语言的运行与调试	9	2.7.5 逗号运算符和逗号表达式	35
1.5.1 C程序的执行过程	9	2.7.6 关系运算符和关系表达式	35
1.5.2 C程序开发的实验方法	9	2.7.7 逻辑运算符和逻辑表达式	36
1.5.3 Visual C++ 6.0 编写程序与调试程序的方法	9	2.7.8 条件运算符和条件表达式	38
习题1	14	2.7.9 位运算符和位运算	39
第2章 数据类型、运算符与表达式	15	习题2	40
2.1 C语言的数据类型	15	第3章 顺序结构程序设计与编译预处理	42
2.1.1 基本数据类型简介	15	3.1 C语句概述	42
2.1.2 基本数据类型在两种环境下的表示 (主要是所占字节数及取值范围)	16	3.2 赋值语句	44
2.2 标识符	16	3.3 赋值表达式与赋值语句的区别	45
2.2.1 可用于标识符的字符集	16	3.4 输入与输出	46
2.2.2 标识符的分类	17	3.4.1 字符数据的输入与输出	46
2.3 常量	17	3.4.2 格式输入与输出	49
2.3.1 整型常量	17	3.4.3 文件操作中的输入与输出简介	59
2.3.2 实型常量	19	3.5 常用函数举例	59
2.3.3 字符常量	20	3.6 编译预处理	60
2.3.4 字符串常量	21	3.6.1 宏定义	60
2.3.5 符号常量与预处理简介	22	3.6.2 文件包含	62
2.4 变量	22	3.7 顺序结构程序设计举例	63
		习题3	64
		第4章 选择结构程序设计	65

4.1 关系运算与逻辑运算	65	第 7 章 函数	134
4.1.1 关系表达式的应用	65	7.1 概述	134
4.1.2 逻辑表达式的应用	65	7.2 函数的定义	134
4.2 if 语句	67	7.3 函数的参数和返回值	135
4.2.1 if 语句的 3 种形式	67	7.3.1 形式参数和实际参数	135
4.2.2 if 语句的嵌套与 {} 语句块的使用	70	7.3.2 函数的返回值	136
4.3 switch 语句	73	7.4 函数的调用	137
4.4 应用举例	76	7.4.1 函数调用的形式和方式	137
习题 4	83	7.4.2 被调函数的声明	138
第 5 章 循环结构程序设计	84	7.4.3 嵌套调用	139
5.1 goto 语句及用 goto 语句构成的循环	84	7.4.4 递归调用	140
5.2 while 语句	85	7.4.5 程序举例	143
5.3 do...while 语句	89	7.5 数组作函数参数	144
5.4 for 语句	92	7.5.1 数组元素作函数实参	144
5.5 几种循环的比较	97	7.5.2 数组名作函数参数	145
5.6 循环嵌套	98	7.5.3 多维数组名作函数实参	147
5.7 break 语句与循环的终止	100	7.6 变量的作用域与存储类别	148
5.8 continue 语句	102	7.6.1 变量的作用域	148
5.9 应用举例	103	7.6.2 变量的存储类别	150
习题 5	114	7.7 内部函数和外部函数	152
第 6 章 数组	116	7.7.1 内部函数	152
6.1 一维数组	116	7.7.2 外部函数	152
6.1.1 一维数组的定义	116	7.8 函数应用举例	152
6.1.2 一维数组元素的引用	117	习题 7	154
6.1.3 一维数组的初始化	117	第 8 章 指针	155
6.1.4 一维数组程序举例	118	8.1 地址和指针的概念	155
6.2 二维数组	121	8.2 变量的指针和指向变量的指针变量	156
6.2.1 二维数组的定义与存储	121	8.2.1 指针变量的定义	157
6.2.2 二维数组元素的引用	122	8.2.2 指针变量的基本运算	157
6.2.3 二维数组的初始化	123	8.2.3 用指针变量作为函数参数	162
6.2.4 二维数组程序举例	123	8.3 数组的指针和指向数组的指针变量	165
6.3 字符数组	125	8.3.1 指向数组元素的指针变量	165
6.3.1 字符数组的定义	125	8.3.2 通过指针引用数组元素	166
6.3.2 字符数组的初始化	125	8.3.3 用数组名作为函数参数	170
6.3.3 字符串和字符串结束标志	125	8.3.4 指向多维数组的指针和指针变量	175
6.3.4 字符数组的输入与输出	126	8.4 字符串的指针和指向字符串的指针变量	180
6.3.5 字符串处理函数	128	8.4.1 字符串的表示形式	180
6.3.6 字符数组程序举例	130	8.4.2 对使用字符指针变量与字符 数组的讨论	181
习题 6	132		

8.4.3 用字符指针作为函数参数	183	9.4 用指针处理链表	213
8.5 函数的指针和指向函数的指针变量	185	9.4.1 链表概述	213
8.5.1 用函数指针变量调用函数	185	9.4.2 处理动态链表所需的函数	215
8.5.2 用指向函数的指针变量作为 函数参数	187	9.4.3 动态链表的基本操作	216
8.6 返回指针值的函数	189	9.5 共用体	220
8.7 指针数组和指向指针的指针	191	9.5.1 共用体的概念及特征	221
8.7.1 指针数组的概念	191	9.5.2 共用体变量的引用方式	222
8.7.2 指向指针的指针	194	9.6 枚举类型	224
8.7.3 用指针数组作为函数参数	195	9.7 用 typedef 定义类型	226
8.8 应用举例	198	9.8 应用举例	227
习题 8	200	习题 9	229
第 9 章 结构体与共用体	201	第 10 章 文件	231
9.1 结构体	201	10.1 文件概述	231
9.1.1 结构体类型的定义	201	10.2 文件类型指针	232
9.1.2 结构体类型变量的定义	202	10.3 标准文件操作	233
9.1.3 结构体变量的引用	204	10.3.1 文件的打开和关闭	233
9.1.4 结构体变量的初始化	204	10.3.2 文件的读/写	234
9.2 结构体数组	206	10.3.3 文件的定位	242
9.2.1 定义结构体数组	206	10.4 文件状态检测函数	243
9.2.2 结构体数组的初始化	206	10.5 文件操作应用举例	243
9.2.3 结构体数组应用举例	207	习题 10	246
9.3 指向结构体类型数据的指针	208	附录 A 常用 ASCII 表	247
9.3.1 指向结构体变量的指针	208	附录 B C 语言中的关键字	248
9.3.2 指向结构体数组的指针	209	附录 C 运算符和结合性	249
9.3.3 用结构体变量和指向结构体的 指针作为函数参数	210	附录 D C 库函数	250
		参考文献	256

第 1 章 程序设计基础

1.1 程序设计语言的发展

机器语言：计算机可以直接识别、执行的二进制代码。如 Z-80 机“加寄存器 B 以累加器”指令为 1000 0000。其优点是执行速度快，但难写、难读、难理解和难修改。

汇编语言：使用助记符来代替机器语言中复杂、繁琐的指令的语言。如 Z-80 机中加法指令为 ADD。其特点是过分依赖于机器码，即仅适合某类相同或相似的计算机，因此不易移植。

高级语言：独立于计算机硬件结构，让用户使用面向问题的形式，更加方便编程人员的语言。高级语言分为如下两类

(1) 编译型：“先编译后执行”，过程为：源代码→目标代码→机器代码，其执行与源程序相互独立。比如 PASCAL，C 等。

(2) 解释型：“边解释边运行”，程序执行离不开源代码的支持，不利于源程序的保护和保密。比如 BASIC，FoxBASE+等。

面向结构（面向过程）的设计语言有：

BASIC（1964 年），FORTRAN（1956 年），COBOL（1960 年）：

面向事务处理的高级语言，适合于编写管理信息系统方面的程序有：

PASCAL（1971 年），C 语言（1974 年），PROLOG（1970 年），FoxBASE+（1986 年）等。

面向对象（Object-Orientation）和可视化（Visual）语言：

C++（1983 年），Java，Visual BASIC，Visual C++，Visual Foxpro，Visual J++，Delphi，PowerBuilder 等。

网页（Web）/网络（Network）编程语言：

HTML，DHTML，JavaScript/Jscript，VBScript，JavaApplet，Java 等。

1.2 算法与程序设计

1.2.1 算法及其表示

“算法”的概念出现在公元 825 年左右，但算法思想在 3500~5000 年前就诞生了。算法是计算机科学中的重要概念之一，它指明了问题的计算过程，是对给定问题解题方案的准确而完整的描述。

计算机没有软件的支持将无法工作，软件是计算机的灵魂，而软件的核心是算法。用计算机解决问题的方法和步骤就是算法，计算机执行算法，用计算机解决各种数据处理问题的主要方式就是寻找和设计各种算法，将算法变成代码的过程即程序设计。

程序员编制的程序包含两个方面的内容：数据结构和算法。数据结构中指定数据的类型和数据的组织形式，而在算法中指定解决问题的操作步骤和描述。

1.2.1.1 算法的特征

算法具备如下 5 个特征：

(1) 有穷性。算法的“有穷性”是指某个算法在有限的操作步骤和用户可接受的时间内完成其执行过程。

(2) 确定性。算法的每一步操作都是含义确切且具体可行的，对于相同的输入则有确定的且类似的输出结果。

(3) 输入。算法可以接受零个或多个输入，允许算法从外界取得相应的数据。

(4) 输出。算法执行后有一个或多个输出。

(5) 可行性。算法中的操作可以通过已经实现的基本操作执行有限次来完成。

1.2.1.2 算法的表示

算法可以用多种方法描述，常见的表示方法有：自然语言描述法、伪代码表示法、传统流程图表示法、N-S 流程图表示法等。

1. 使用自然语言表示算法

自然语言就是人们常用的语言，可以是汉语、英语或其他语言。

用自然语言表示通俗易懂，但因为描述不够确切，容易出现“歧义”，而且，使用自然语言描述分支结构和循环结构比较累赘，不很方便。用户在算法不够清晰具体的时候，使用自然语言描述算法可以帮助自己理清思路。

【例 1-1】已知杯 A 中盛放的是油，杯 B 中盛放的是水，另有一个空杯 C，三个杯容积相同，请用自然语言表示算法。

【解答】

第 1 步：将 A 中的油全部倒入 C 中，使 A 成为空杯。

第 2 步：将 B 中的水全部倒入 A 中，使 B 成为空杯。

第 3 步：将 C 中原在 A 中的油，倒入 B 中，完成 A 与 B 中液体的交换。

【例 1-2】描述计算并输出 $z=y/x$ 的流程。

【解答】

第 1 步：输入 x, y 。

第 2 步：判断 x 是否为 0：

若 $x=0$ ，则输出错误信息；

否则计算 $y/x \Rightarrow z$ ，且输出 z 。

上述两个示例，都是比较简单的问题，已经占用了相当的文字篇幅，如果问题稍复杂一些，需要更多的文字才可以描述清楚。

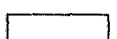
对于同一个问题可以有不同的解题方法和步骤，例 1-1 还有其他的算法。同一个问题的解法有优劣之分，有些算法简捷，容易实现，有些算法冗长，执行的效率比较低，因此，在算法的选取和设计中，要考虑算法的质量。

2. 使用传统流程图表示算法

传统流程图用规定的一组图形符号和文字说明来表示各种操作的算法，如表 1-1 所示。

表 1-1 中符号由美国国家标准化协会 ANSI (American National Standard Institute) 规定，用流程图表示算法，直观形象，易于理解，但画起来比较困难，由于绘制出的流程图幅面偏大，造成绘制和修改比较困难。

表 1-1 传统流程图常用符号

符号	符号名称	含义
	起止框	表示算法的开始和结束
	输入/输出框	表示输入/输出操作
	判断框	表示对框内的条件进行判断
	处理框	表示对框内的内容进行处理
	流程线	流程的方向
	连接点	避免流程线的交叉或过长，使流程图清晰

【例 1-3】用传统流程图表示对两个数按从小到大的顺序输出的算法。

【解答】传统流程图如图 1-1 所示。

在结构化程序设计中，流程图只包括 3 种基本结构：

(1) 顺序结构。顺序结构是结构化程序设计中最简单的一种结构，这种结构有一个入口和一个出口，中间的若干操作按照顺序依次执行，如图 1-2 所示，先执行操作 A，然后执行操作 B。

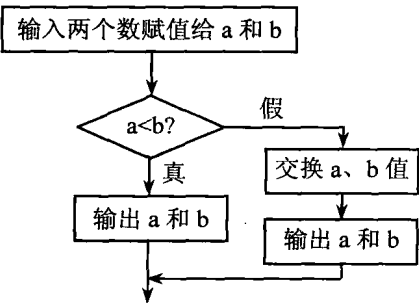


图 1-1 由小到大输出两个数

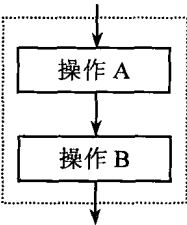


图 1-2 顺序结构

(2) 选择结构。选择结构与分支结构属同一结构，它由一个条件和两组语句组成，计算机根据条件的真假来选择执行的分支，如果判断成立则执行操作 A，否则执行操作 B。选择结构的流程图如图 1-3 所示。

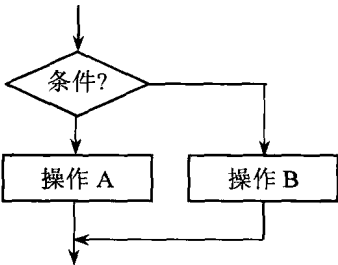


图 1-3 选择结构

（3）循环结构。循环结构是指重复执行同一组程序段的结构。循环结构由两部分组成，其一是循环条件；其二是循环体，被重复执行的程序段称为循环体。循环体能否继续执行由循环条件决定。根据循环条件出现的位置，循环结构又分为“当型循环”和“直到型循环”两类。

“当型循环”中，当逻辑条件成立时，反复执行循环体，直到逻辑条件不成立时结束循环执行，如图 1-4 所示。

“直到型循环”中，反复执行循环体，直到循环条件不成立时结束循环，如图 1-5 所示。

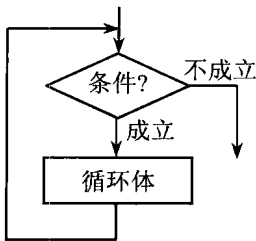


图 1-4 “当型循环”结构

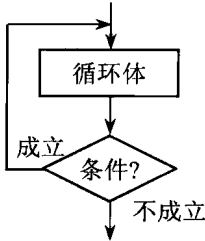


图 1-5 “直到型循环”结构

3. 使用 N-S 流程图表示算法

由于传统流程图存在不足，提出一种新的结构化流程图形式，这种形式的流程图的主要特点是取消了流程线，不再显示控制流，整个算法的流程写在一个矩形框内，这种形式的流程图称为 N-S 流程图。根据结构化程序的 3 种基本结构，N-S 流程图有 3 种基本形式，其中循环结构 N-S 流程图又分为“当型循环”和“直到型循环”两种。N-S 流程图表示的 3 种基本结构如下所示：

- （1）顺序结构的 N-S 流程图如图 1-6 所示。
- （2）选择结构的 N-S 流程图如图 1-7 所示。
- （3）循环结构。“当型循环”结构的 N-S 流程图如 1-8 所示，“直到型循环”结构的 N-S 流程图如图 1-9 所示。

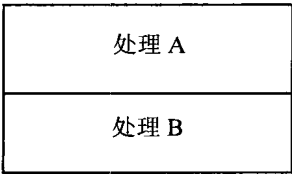


图 1-6 顺序结构

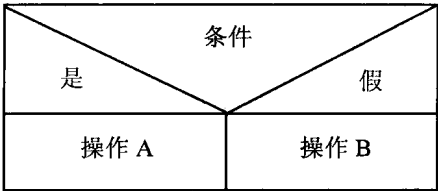


图 1-7 选择结构

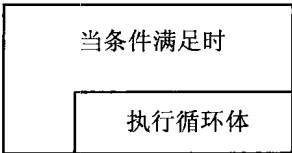


图 1-8 “当型循环”结构

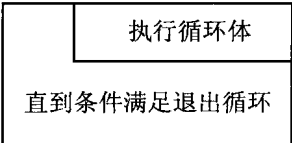


图 1-9 “直到型循环”结构

4. 使用伪代码表示算法

伪代码使用介于自然语言和计算机语言之间的文字和符号来描述算法。这种表示方式比

较接近于计算机语言，如果结合相应的语法，即可以修改成为程序。

【例 1-4】使用伪代码描述数据 A 与数据 B 进行交换的算法，要求不引入临时数据。

【解答】使用伪代码描述如下：

- (1) $A=A+B$ ，将 A 与 B 相加赋值给 A，则 A 为原数据 A+B 的和；
- (2) $B=A-B$ ，原两数和减去 B 即 A 的值，现赋值给 B；
- (3) $A=A-B$ ，原两数和减去最近更新的 B 即 A 的新值（原数据 B）；
- (4) 两数 A、B 完成交换。

由于伪代码不能直接执行，但表示算法方便，容易理解。伪码的表现形式灵活自由，格式紧凑，但没有严谨的语法格式。

1.2.2 结构化程序设计 with 面向对象程序设计

编写高质量的程序，除了熟练掌握程序设计语言语法外，还要研究程序设计方法和设计技巧。在不断练习和思考中深入理解程序设计的内涵，才可以逐步提高程序设计的能力。

目前，软件开发行业中有两种典型的程序设计方法，一种为结构化程序设计方法；另一种为面向对象的程序设计方法。

“结构化程序设计”方法。传统的“结构化程序设计”（Structured Programming）最早由 E.W.Dijkstra 在 1965 年提出的，它的主要观点是采用自顶向下、逐步求精的程序设计方法，使用 3 种基本结构构造程序，任何程序都可由顺序、选择、循环 3 种基本控制结构编写出来。

大型软件的开发，需要科学管理，缺乏规划的软件开发和不完善的管理，会导致软件开发的进度被推迟，成本超过预算等问题。“结构化程序设计”方法旨在提高程序的可读性，保证软件的质量，降低软件成本，提高软件开发和维护的效率。因此，软件的开发应当按照工程的生产方式来组织和管理，每个成员都必须按照统一的规划、方法进行工作，使生产的软件有统一的标准和风格，以便于生产、推广和维护。

“面向对象程序设计（OOP）”方法。所谓面向对象的程序设计，就是把面向对象的思想应用到软件工程中，并指导开发维护软件。

对象是由数据和容许的操作组成的封装体，所谓面向对象，就是基于对象的概念，以对象为中心，类和继承为构造机制，认识了解刻画客观世界以及开发出相应的软件系统。

“面向对象程序设计”就是通过数据抽象，将许多实例中共性的数据和为操作这些数据所需要的算法抽取出来，并进行封装和数据隐藏，形成一个新的数据类型——“类”类型。

面向对象程序设计的关键在于如何进行数据的抽象和封装，以及对处理数据的过程的设计。面向对象程序设计的最大优点是允许“继承”，即在某个类的基础上可以派生出新类。目前的面向对象程序设计开发工具都提供了大量的类，用户可以直接使用这些类，或通过对这些类的扩充和重用形成新的类。与其他数据类型一样，只有类型对应的变量才是实际操作的对象。而类的变量我们称为“对象”。对“对象”的操作就是利用对象进行程序设计，也就是面向对象程序设计。

初学者在学习编程的时候应以“结构化程序设计”为基础，“面向对象”编程中涉及到的编程思想和某些经典算法仍然来自“结构化程序设计”。

1.3 C 语言简介

1.3.1 C 语言的产生与发展

C 语言是在 20 世纪 70 年代初问世，是目前国际上广泛流行的一种结构化的程序设计语言。1978 年由美国电话电报公司开始了 C 语言的最初研发，它不仅是开发系统软件很好的工具，而且也是开发应用软件很好的程序设计语言。

当初的 C 语言是为开发 UNIX 操作系统而研制的，1983 年美国国家标准学会（ANSI）将 C 语言标准化，随着 C 语言的广泛应用又不断推出新的 C 语言版本，其性能也越来越强。到了 1975 年 UNIX（第 6 版）的推出和随着面向对象程序设计（OOP）技术的出现，C 语言的突出优点就引起了人们的普遍关注，ANSI C 标准于 1989 年被采用，该标准定义为 C89，到了 1995 年，出现了 C 的修订版，其中增加了一些库函数，出现了初步的 C++，在此基础上，C89 成为 C++ 的子集。此后，C 语言不断发展，在 1999 年又推出了 C99，C99 在基本保留了 C 的特性的基础上增加了一系列新的特性，随后又几经修改和完善，它也从面向过程的编程语言发展到面向对象的程序设计语言，目前可在微机上运行的 C 语言版本主要有 Turbo C、Quick C、Visual C++ 等版本。

1.3.2 C 语言的特点与应用领域

C 语言的主要特点有：

- （1）语言简洁，紧凑，使用方便，灵活。
- （2）运算符丰富。
- （3）数据结构丰富，具有现代化语言的各种数据结构。
- （4）具有现代化的控制语句（如 if...else 语句，while 语句，do...while 语句，switch 语句，for 语句）。
- （5）语法限制不太严格，程序设计自由度大。
- （6）C 语言能进行“位”（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。
- （7）生成目标代码质量高，程序执行效率高。
- （8）程序可移植性好。
- （9）绘图能力强，可完成图形图像相关的程序设计。

1.4 C 语言的结构与执行

1.4.1 C 语言源程序的结构特点

C 语言程序的一般形式如下：

```
编译预处理部分
全局变量定义
main() /*主函数*/
{
```

```

    全局定义序列
语句序列
}
func1() /*自定义函数 func1*/
{
    .....
}
func2() /*自定义函数 func2*/
{
    .....
}

```

C 语言源程序的结构特点：

- (1) 一个 C 语言源程序可以由一个或多个源文件组成。
- (2) 每个源文件可由一个或多个函数组成。
- (3) 一个源程序不论由多少个文件组成，都有一个且只能有一个 `main()` 函数，即主函数。
- (4) 源程序中可以有预处理命令，预处理命令通常应放在源文件或源程序的最前面。
- (5) 每一个声明，每一条语句都必须以分号结尾。但预处理命令，函数头和花括号 “}” 之后不能加分号。
- (6) 标识符，关键字之间必须至少加一个空格以示间隔。
- (7) 可以用 `/*.....*/` 对 C 语言程序中的任何部分进行注释。一个好的、有使用价值的源程序都应当加上必要的注释，增加程序的可读性和可维护性。

1.4.2 高级语言程序的执行过程

高级语言与具体计算机无关，是一种能方便描述算法过程的计算机程序设计语言。高级语言种类千差万别，但一般包含有以下 4 种成分：数据、运算、控制和传输。“数据”成分用来描述程序所涉及的数据；“运算”成分用来描述运算；“控制”成分用来表达程序的控制构造；“传输”成分用来表达数据的传输。由于高级语言程序主要是描述计算机的解题过程，即描述复杂的加工处理过程，所以也称这种高级语言为“面向过程语言”。

用高级语言编写的程序称为“源程序”。计算机不能直接执行源程序的各语句，通常有“解释”和“编译”两种方法处理源程序。“解释”方式，即让计算机运行解释程序，解释程序逐句取出源程序中的语句，对它作解释执行，输入数据，产生结果。“编译”方式，即先运行编译程序，从源程序中一次性翻译产生计算机可直接执行的二进制程序（称为目标程序），然后让计算机将目标程序结合具体的计算机硬件产生相应的“可执行”程序，输入数据，产生结果。

解释方式的主要优点是计算机与人的交互性好，调试程序时，能一边执行一边直接改错，能较快得到一个正确的程序。缺点是逐句解释执行，运行速度慢。

编译方式的主要优点是计算机运行目标程序快，缺点是修改源程序后必须重新编译以产生新的目标程序。

1.4.3 程序设计的开发期与开发习惯

程序设计的过程可以分为若干个相互关联的阶段。针对问题的要求，从分析问题的需求出发，逐步深入，到最后编制出能正确编译并可执行的程序。

第 1 阶段，分析问题，确定问题的需求。

接受任务后，首先要深入了解待解决问题，深刻掌握题意，知道要“做什么”。

第2阶段，分析问题，建立数学模型。

使用数学方法来描述实际问题的方法称为建立数学模型。建立数学模型的过程实质上就是通过一系列分析和实验找出待解决问题中涉及到的运算操作和活动的规律，然后进行归纳，并作抽象的数学描述。成功建立数学模型，找出规律离待解问题更近一步。

第3阶段，选择计算方法。

对于同一个数学模型，存在多种解决方案，如例 1-1 和例 1-4 中关于两数交换就有两种处理方法，再如计算定积分的计算机求解，可以采用矩形法，也可以采用梯形法。虽然各种不同的方法均可解决问题，但在计算的精度，处理的速度和需要占用的资源都存在差异。因此，要针对具体问题，选择合适的计算方法以达到更好的处理效果。

第4阶段，设计算法，绘制流程图。

根据所选择的数学模型设计可行的计算方法，并把计算过程使用形象、直观的流程图清晰反映算法的基本思想和操作步骤。有了流程图，程序的编写工作就显得简单有条理，有利于程序的调试、修改和交流。

第5阶段，代码编写。

编程者要熟悉语言的语义和各种语法规则和规定，将算法准确而具体地表达出来，形成一个可以编译的完整的程序。

第6阶段，调试程序。

所谓调试程序，是将编制的程序投入实际运行前，用手工或编译程序等方法进行测试，修正语法错误和逻辑错误的过程。查错时，可以采取分段调试、逐层分析等有效的调试手段，以期待得到一个完整的能正常工作的程序。

第7阶段，整理资料和交付使用。

程序编写和测试成功后，为了使用户能了解程序的具体功能和掌握程序的运行操作，必须将程序设计阶段形成的资料和相关说明进行归类整理，形成程序说明书，内容包括：程序名称、任务的具体要求、给定的原始数据、算法、程序框图、程序清单、测试及运行结果、程序操作说明、程序运行环境要求和其他资料。

关于程序的开发，从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时应遵循以下规则：

- (1) 程序所采用的算法要尽量简单，符合人们一般的思维方式。
- (2) 标识符的命名尽量采取“见名知义，常用从简”的原则。
- (3) 采用“缩进”的方式书写程序。低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写，以便看起来更加清晰，增加程序的可读性。
- (4) 程序中可用/*……*/或//注释，以提高程序的可读性。
- (5) 在输入数据前，加一条输出语言，提醒用户输入什么样的数据。
- (6) 编译预处理部分的语句，不能加分号。
- (7) 每一行写一条语言。
- (8) 用{}号括起来的部分，通常表示程序的某一层结构。{}一般与该结构语句的第一个字母对齐，并单独占一行。

在编程时应养成良好的编程习惯，这样可以达到事半功倍的效果。

1.5 C 语言的运行与调试

1.5.1 C 程序的执行过程

C 源程序的扩展名为 (*.c)，经过编译程序处理后的源程序生成目标程序 (*.obj)，目标程序经过连接程序处理，生成可执行程序 (*.exe)，可执行程序能够直接运行。整个过程如图 1-10 所示。

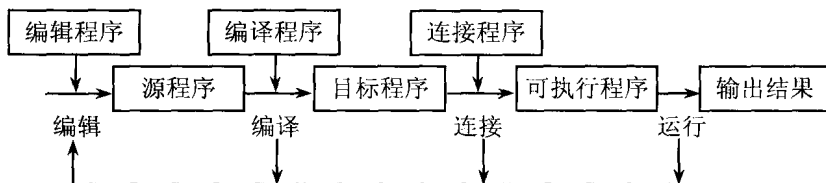


图 1-10 源程序的变化过程

标准 C 程序的执行是从 `main()` 函数开始执行的，如果 `main()` 函数中调用了其他函数，其他函数调用结束后，返回主函数继续执行。

1.5.2 C 程序开发的实验方法

Visual C++ 6.0 是微软公司开发并广泛流行的基于 Windows 平台的 C 和 C++ 语言集成开发环境，该环境具有良好的工程管理模式，适合团队开发大型软件，操作者可以利用它实现 C 语言程序的编辑、编译、连接、运行和调试。Visual C++ 6.0 运行在 Windows 操作系统上，其基本的光标移动、插入、复制、粘贴和删除等功能的操作方法都与普通的文本编辑方法一致。下面介绍在 Visual C++ 6.0 集成开发环境下运行 C 语言程序的一般方法。

开发程序之初，需要建立一个工作区，工作区的主要功能是管理多个工程，在某个工程的管理下，可以编写和调试 C 语言程序。适应该编程环境的程序员将会受益匪浅。

- (1) 安装 Microsoft Visual C++ 6.0，安装完成后不需要配置环境。
- (2) 建立工作区。
- (3) 打开工作区，并在该工作区下建立工程。
- (4) 编写源程序。
- (5) 编译源程序。
- (6) 组建源程序。
- (7) 调试和执行源程序。

1.5.3 Visual C++ 6.0 编写程序与调试程序的方法

以下部分介绍在 Visual C++ 6.0 环境下开发程序的操作步骤：

- (1) 启动 Visual C++ 6.0。

单击“开始”→“程序”→“Microsoft Visual C++6.0”→“Microsoft Visual C++ 6.0”菜单项，启动 Visual C++ 6.0。如果在桌面上已创建 Visual C++ 6.0 的快捷方式，双击该快捷方式的图标，也可启动 Visual C++ 6.0。