

ρ ROGRAMMING IN C

C语言程序设计教程

黄苏雨 吴佳芬 张帆 主编



机械工业出版社
China Machine Press

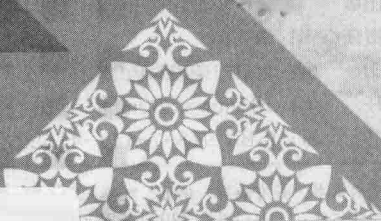
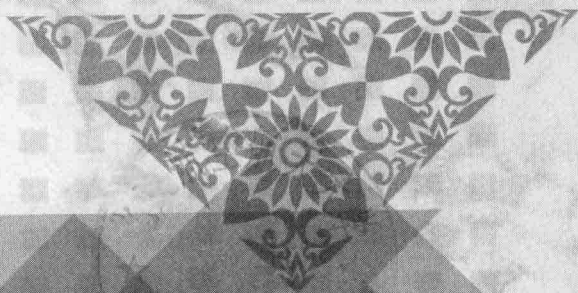
计算机基础课程系列教材

P

ROGRAMMING IN C

C语言程序设计教程

黄苏雨 吴佳芬 张帆 主编



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

C 语言程序设计教程 / 黄苏雨, 吴佳芬, 张帆主编. —北京: 机械工业出版社, 2014.9
(计算机基础课程系列教材)

ISBN 978-7-111-47622-1

I. C… II. ①黄… ②吴… ③张… 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 (2014) 第 185945 号

本书系统地介绍 C 语言的基本知识和程序设计的基本方法。全书共 11 章, 主要内容包括: C 语言的基本知识, 数据类型、运算符和表达式, 基本程序控制结构 (顺序、选择、循环), 数组, 函数, 指针, 结构体和共用体, 文件, 编译预处理。

本书主要作为高等院校非计算机专业学生的程序设计入门教材, 也可作为全国计算机等级考试二级 C 语言程序设计的参考用书。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 朱秀英

责任校对: 董纪丽

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2014 年 10 月第 1 版第 1 次印刷

开 本: 185mm×260mm 1/16

印 张: 19.25

书 号: ISBN 978-7-111-47622-1

定 价: 35.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

前 言

随着科学技术的发展,计算机已深入日常工作和学习的方方面面,极大地加快了社会信息化的进程。对于非计算机专业的理工科大学生来说,已不仅仅局限于会使用计算机,还要掌握一门计算机语言并能使用其进行程序设计,以解决专业领域中的问题。针对非计算机专业学生的特点,并结合多年从事该课程教学的经验,我们编写了本教材。

C语言是使用最广泛的程序设计语言之一,其功能强大、灵活方便、可移植性好,深受广大科技人员和专业编程爱好者的喜爱,一直在软件开发过程中发挥着重要作用。全国计算机等级考试、全国计算机软件著作权及水平考试、全国计算机应用技术证书考试等都将C语言纳入其考试科目,众多高校也将C语言作为程序设计课程的入门语言。

为了与目前全国计算机等级考试使用的版本一致,本书基于Visual C++ 6.0的集成开发环境,理论部分实例均在英文版Visual C++ 6.0编程环境中编译、连接、运行。为方便学生实践,实验部分采用中文版Visual C++ 6.0。本书在编写过程中力求由浅入深、循序渐进、结构清晰、层次分明,选取典型案例进行讲解。本书注重对程序设计语言基本概念、语法规则、程序结构和编程方法进行详述,以期让读者能尽快迈进程序设计的大门。

本书共分为11章,主要包括:

第1章 C语言基础知识。介绍了程序设计的概念、C语言的发展和特点、C程序的结构和开发过程、Visual C++ 6.0集成开发环境。

第2章 数据类型、运算符和表达式。介绍了常量和变量的基本概念、基本类型数据的使用、不同数据类型之间的转换,算术运算符、关系运算符、逻辑运算符、赋值运算符、逗号运算符、位运算符及其表达式。

第3章 顺序结构程序设计。介绍了C语言的语句分类、赋值语句、字符的输入与输出、格式的输入与输出、顺序结构程序设计方法。

第4章 选择结构程序设计。介绍了选择结构中的if语句、条件运算符和条件表达式、switch语句。

第5章 循环结构程序设计。介绍了循环结构中的while语句、do...while语句和for语句,并对三种循环结构进行了比较。

第6章 数组。介绍了一维数组、二维数组和字符数组的定义和存储、初始化、数组元素的引用和输入输出。

第7章 函数。介绍了函数的定义,函数调用的一般形式、函数的参数和函数的返回值,函数的嵌套调用和递归调用,标识符的作用域和存储类别,数组作为函数参数。

第8章 指针。介绍了地址与指针、指针变量的应用、指针与数组、指针与函数。

第9章 结构体和共用体。介绍了结构体和共用体两种数据类型的定义、引用、初始化和赋值,结构体数组,结构体指针,结构体与函数。

第10章 文件。介绍了文件的概念、文件指针、文件处理的基本过程和用于处理文件的函数。

第11章 预处理。介绍了编译预处理的概念和常用的编译预处理命令,包括宏定义、文

件包含和条件编译等。

本书第1章由黄兴鼎编写,第2章由黄兴鼎、陈红编写,第3、4章由张帆编写,第5章由李芬编写,第6章由吴佳芬编写,第7章由黄苏雨编写,第8章由刘飞编写,第9章由周明编写,第10章由李林编写,第11章由赵莉编写。实验1和实验2由黄兴鼎编写,实验3和实验4由张帆编写,实验5由李芬编写,实验6由吴佳芬编写,实验7由黄苏雨编写,实验8由刘飞编写,实验9由周明编写,实验10由李林编写,实验11由赵莉编写。全书由黄苏雨、吴佳芬和张帆统稿。在本书的编写和出版过程中,得到了各级领导和机械工业出版社的大力支持,在此表示衷心的感谢。

为了便于教学,我们为选用本教材的任课教师免费提供电子教案和自测题参考答案,请登录华章网站(www.hzbook.com)免费下载或通过电子邮件(ncuhuangshuyu@sina.com)与我们联系。

由于编者水平所限,教材中难免有疏漏和欠缺之处,敬请广大读者提出宝贵意见。

编者

2014年6月于武汉

教学建议

教学内容	教学要求	授课学时	实验学时
第 1 章 C 语言基础知识	- 明确学习 C 语言的目的 - 了解 C 语言的起源及其标准 - 掌握 C 语言的基本特性、C 语言程序的开发过程和 C 语言编程环境的基本使用	4	2
第 2 章 数据类型、运算符和表达式	- 掌握基本数据类型 - 掌握变量和常量的概念 - 掌握运算符和表达式的运用	4	4
第 3 章 顺序结构程序设计	- 掌握基本输入输出函数的使用 - 掌握顺序结构程序设计	2	2
第 4 章 选择结构程序设计	- 掌握 if 语句的基本形式和使用 - 掌握用 switch 语句实现多分支选择结构	2	2
第 5 章 循环结构程序设计	- 理解 break 与 continue 语句的使用及区别 - 掌握循环结构（while 语句、do...while 语句、for 语句）的使用	4	4
第 6 章 数组	- 掌握一维数组的定义、初始化及数组元素的引用 - 了解二维数组的定义、初始化、使用 - 掌握字符数组的定义和使用	4	4
第 7 章 函数	- 掌握函数的定义和调用 - 掌握函数的嵌套调用和递归调用 - 理解变量的作用域和存储类别	4	4
第 8 章 指针	- 了解指针和数组的关系 - 掌握指针的概念和定义方法 - 掌握指针变量的简单应用 - 掌握数组名作为函数参数	4	4
第 9 章 结构体和共用体	- 掌握结构体类型的定义、结构体变量的定义和使用以及结构体数组的使用 - 了解共用体类型的定义、初始化及使用 - 掌握链表的建立、遍历、删除与插入操作	4	4
第 10 章 文件	- 了解文本文件和二进制文件的概念 - 掌握文件输入输出库函数的使用	2	2
第 11 章 预处理	- 掌握不带参数的宏定义及其使用 - 了解带参数的宏定义 - 掌握“文件包含”处理的概念及其作用	2	2
合计		36	34

推荐阅读

参考文献



大学计算机基础

作者：陈明 王锁柱 主编 ISBN: 978-7-111-43767-3 定价：35.00元



数据库与数据处理：Access 2010实现

作者：张玉洁 孟祥武 编著 ISBN: 978-7-111-40611-2 定价：35.00元



数据库原理及应用

作者：王丽艳 郑先锋 刘亮 编著 ISBN: 978-7-111-40997-7 定价：33.00元



计算机网络教程 第2版

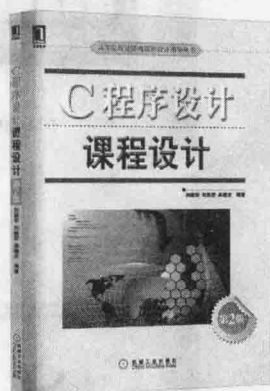
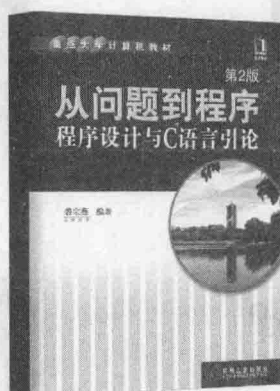
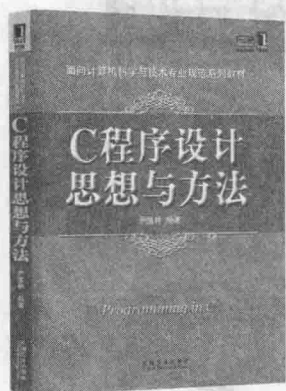
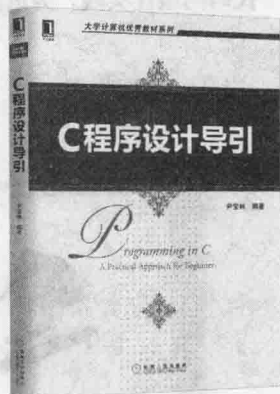
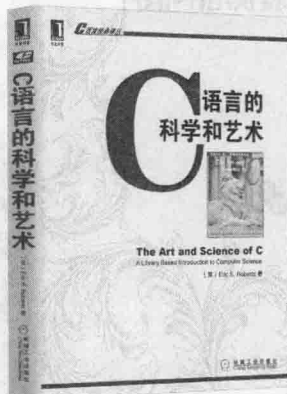
作者：熊建强 黄文斌 彭庆喜 主编 ISBN: 978-7-111-38804-3 定价：39.00元



Access 2010数据库程序设计教程

作者：熊建强 吴保珍 黄文斌 主编 ISBN: 978-7-111-43681-2 定价：39.00元

推荐阅读



C程序设计语言（第2版·新版）

作者: Brian W. Kernighan 等 ISBN: 978-7-111-12806-0 定价: 30.00元

C语言的科学和艺术

作者: Eric S. Roberts ISBN: 978-7-111-34775-0 定价: 79.00元

C程序设计导引

作者: 尹宝林 ISBN: 978-7-111-41891-7 定价: 35.00元

C程序设计思想与方法

作者: 尹宝林 ISBN: 978-7-111-25495-9 定价: 36.00元

从问题到程序——程序设计与C语言引论 第2版

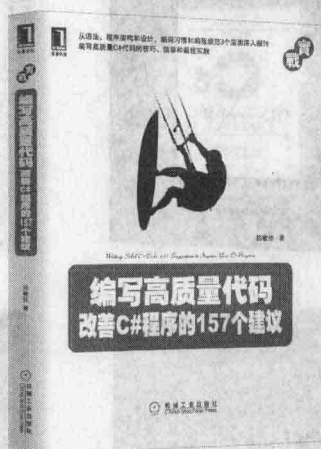
作者: 裴宗燕 ISBN: 978-7-111-33715-7 定价: 39.00元

C程序设计课程设计 第2版

作者: 刘振安 等 ISBN: 978-7-111-28541-0 定价: 24.00元

推荐阅读

从语法、应用架构、工具、框架、编码风格和编程思想等多角度探讨编写高质量代码的最佳实践，为进阶修炼提供绝佳指导！



目 录

前言

教学建议

第1章 C语言基础知识	1	2.2.5 符号常量	21
1.1 程序设计的基本概念	1	2.3 整型数据	22
1.1.1 程序和程序设计	1	2.3.1 整型常量	22
1.1.2 算法的概念和描述	2	2.3.2 整型变量	23
1.1.3 结构化程序设计	5	2.3.3 枚举型数据	26
1.2 C语言概述	5	2.4 实型数据	26
1.2.1 C语言的发展历史及以C语言为基础的 其他语言	6	2.4.1 实型常量	26
1.2.2 C语言的标准与版本	6	2.4.2 实型变量	28
1.2.3 C语言的特点	6	2.5 字符型数据	30
1.2.4 C语言的市场占有率与为什么 学习C语言	7	2.5.1 字符型常量	30
1.3 C语言程序的组成	8	2.5.2 字符型变量	31
1.3.1 简单的C程序	8	2.5.3 字符串常量	33
1.3.2 C程序的结构	9	2.6 不同数据类型之间的转换	33
1.3.3 C程序的开发过程	10	2.6.1 自动类型转换	33
1.4 Visual C++ 6.0 集成开发环境	11	2.6.2 强制类型转换	33
1.4.1 Visual C++ 6.0 的启动与退出	11	2.6.3 求字节数运算符 sizeof	34
1.4.2 Visual C++ 6.0 集成开发环境 概述	12	2.7 算术运算符	35
1.4.3 程序调试和测试	13	2.7.1 算术运算符和算术表达式	35
本章小结	15	2.7.2 自增自减运算符和表达式	37
自测题	16	2.8 关系运算符、逻辑运算符及其 表达式	38
第2章 数据类型、运算符和表达式	19	2.8.1 关系运算符和关系表达式	38
2.1 数据类型	19	2.8.2 逻辑运算符和逻辑表达式	39
2.2 常量、变量和标识符	20	2.9 赋值运算符、逗号运算符及其 表达式	40
2.2.1 字符集	20	2.9.1 简单赋值运算符和简单赋值 表达式	40
2.2.2 标识符	20	2.9.2 复合赋值运算符与复合赋值 表达式	41
2.2.3 关键字	21	2.9.3 逗号运算符和逗号表达式	41
2.2.4 常量与变量	21	2.10 位运算符	42
		2.10.1 位逻辑运算符	43
		2.10.2 移位运算符	45

2.10.3 位复合赋值运算符	45	5.7 循环结构程序设计举例	92
本章小结	46	5.8 三种循环语句的比较	94
自测题	46	本章小结	94
第3章 顺序结构程序设计	49	自测题	95
3.1 C语言语句	49	第6章 数组	101
3.2 赋值语句	50	6.1 数组概述	101
3.3 字符的输入与输出	51	6.2 一维数组	101
3.3.1 输入与输出函数的头文件	51	6.2.1 一维数组的定义和存储	101
3.3.2 字符输入函数 getchar() 与字符 输出函数 putchar()	52	6.2.2 一维数组的初始化	102
3.4 格式的输入与输出	53	6.2.3 一维数组元素的引用	103
3.4.1 格式输入函数 scanf()	53	6.2.4 一维数组元素的输入输出	105
3.4.2 格式输出函数 printf()	56	6.3 二维数组	107
3.5 顺序结构	58	6.3.1 二维数组的定义和存储	108
3.5.1 顺序结构的特点	59	6.3.2 二维数组的初始化	109
3.5.2 顺序结构程序设计举例	59	6.3.3 二维数组元素的引用	110
本章小结	60	6.3.4 二维数组元素的输入输出	111
自测题	60	6.4 字符数组	113
第4章 选择结构程序设计	63	6.4.1 字符数组的定义	113
4.1 条件语句	63	6.4.2 字符数组的初始化	113
4.1.1 if语句	63	6.4.3 字符数组的引用	114
4.1.2 if...else语句	65	6.4.4 字符串与字符串结束标志	115
4.1.3 if...else if语句	66	6.4.5 字符数组的输入输出	117
4.1.4 if语句的嵌套	69	6.4.6 字符串处理函数	121
4.2 条件运算符和条件表达式	70	本章小结	127
4.3 switch语句	72	自测题	127
4.4 选择结构程序设计举例	74	第7章 函数	133
本章小结	76	7.1 函数概述	133
自测题	77	7.2 函数的定义	134
第5章 循环结构程序设计	81	7.3 函数调用	136
5.1 概述	81	7.3.1 函数的调用	136
5.2 while循环	81	7.3.2 函数的参数	138
5.3 do...while循环	84	7.3.3 对被调函数的声明	139
5.4 for循环	86	7.3.4 函数的返回值	141
5.5 break语句和continue语句	88	7.4 函数的嵌套调用和递归调用	142
5.5.1 break语句	88	7.4.1 函数的嵌套调用	142
5.5.2 continue语句	90	7.4.2 函数的递归调用	143
5.6 循环的嵌套	90	7.5 标识符的作用域和存储类别	146
		7.5.1 变量的作用域	146

7.5.2 变量的存储类别	149	9.5.3 传递指向结构体变量的指针	201
7.5.3 函数的存储类别	154	9.6 共用体	202
7.6 数组作为函数参数	155	9.6.1 定义共用体类型	202
7.6.1 数组元素作为函数参数	155	9.6.2 定义共用体变量	203
7.6.2 数组名作为函数参数	156	9.6.3 共用体和结构体的比较	204
本章小结	157	9.6.4 共用体类型的指针	205
自测题	158	9.6.5 共用体变量成员的引用	205
第8章 指针	165	9.7 链表	207
8.1 指针与指针变量的概念	165	9.7.1 自引用结构	207
8.1.1 地址与指针	165	9.7.2 链表结构	207
8.1.2 指针变量	166	9.7.3 创建与遍历链表	208
8.2 指针变量的应用	166	9.7.4 删除链表结点	208
8.2.1 定义指针变量	166	9.7.5 插入链表结点	210
8.2.2 指针运算符	166	本章小结	211
8.2.3 指针变量的初始化	168	自测题	212
8.2.4 把指针作为函数参数传递	168	第10章 文件	217
8.3 指针与数组	169	10.1 概述	217
8.3.1 指向数组的指针	170	10.1.1 文件定义及分类	217
8.3.2 指针的算术运算	175	10.1.2 文件操作过程	218
8.3.3 指针的关系运算	177	10.1.3 文件缓冲区	219
8.3.4 把数组名作为函数参数传递	179	10.1.4 文件类型指针	219
8.4 指针与函数	180	10.2 打开和关闭文件	220
8.4.1 返回指针的函数	180	10.2.1 打开文件	220
8.4.2 函数指针	181	10.2.2 关闭文件	221
本章小结	183	10.3 顺序读写文件	222
自测题	184	10.3.1 文件位置指针	222
第9章 结构体和共用体	191	10.3.2 判断文件结束函数	223
9.1 定义结构体类型	191	10.3.3 文件读写函数	223
9.2 定义和使用结构体变量	192	10.4 随机读写文件	230
9.2.1 结构体变量的定义	192	10.4.1 rewind() 函数	230
9.2.2 结构体变量的引用	194	10.4.2 fseek() 函数	230
9.2.3 结构体变量的初始化	194	10.4.3 ftell() 函数	230
9.2.4 结构体变量的赋值	195	10.5 文件出错处理	232
9.3 结构体数组	196	10.5.1 ferror() 函数	232
9.4 结构体指针	198	10.5.2 clearerr() 函数	233
9.5 结构体与函数	200	本章小结	233
9.5.1 传递结构体变量的成员	200	自测题	233
9.5.2 传递结构体变量	201		

第 11 章 预处理	237	实验 4 选择结构程序设计	260
11.1 宏定义	237	实验 5 循环结构程序设计	263
11.1.1 不带参数的宏定义	237	实验 6 数组	266
11.1.2 带参数的宏定义	241	实验 7 函数	271
11.2 文件包含	243	实验 8 指针	276
11.3 条件编译	246	实验 9 结构体与共用体	280
本章小结	248	实验 10 文件	282
自测题	249	实验 11 预处理	286
实验	252	附录 A ASCII 码表	288
实验 1 C 语言基础知识	252	附录 B 运算符及其优先级和结合性	289
实验 2 数据类型、运算符和表达式	256	附录 C 常用标准库函数	290
实验 3 顺序结构程序设计	259	参考文献	295

第 1 章 C 语言基础知识

C 语言是一种通用的编程语言，深受广大科技工作者喜爱。本章主要介绍程序设计和算法的基本概念、C 语言的发展历史、C 语言的标准和版本、C 语言的特点、为什么学习 C 语言，以及 C 语言程序的组成、结构、编写格式和 C 语言程序的开发过程，最后介绍 Visual C++ 6.0 集成环境下的上机操作过程和调试方法。学习本章的目的是使读者对 C 语言和程序设计有一个大概的了解，并掌握运行简单程序的操作步骤。

1.1 程序设计的基本概念

本节介绍程序、程序设计、算法和结构化程序设计的概念。

1.1.1 程序和程序设计

1. 程序

程序是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。计算机程序是软件开发人员按用户需求、用程序设计语言开发的、为实现预期目的、适合计算机执行的指令（语句）序列的集合。程序执行的效率与数据的存储结构密切相关，计算机程序分为系统程序和应用程序两大类。

2. 程序设计语言

程序设计语言是编写程序的工具，是软件系统的重要组成部分，又称为计算机语言。程序设计语言可以分为机器语言、汇编语言和高级语言三大类。

(1) 机器语言

计算机只懂得机器语言，机器语言是由 0 和 1 组成的系列代码，所编辑的程序称为机器语言程序，计算机可以直接执行，它是面向机器的程序设计语言。

(2) 汇编语言

汇编语言是用助记符代替机器语言的二进制操作码，用地址符号或标号代替地址码，把机器语言变成汇编语言。

例如，实现 $1+1$ 汇编语言要写成

```
MOV AX,1  
ADD AX,1
```

使用汇编语言编写的程序，机器不能直接识别，要由一种程序将汇编语言翻译成机器语言，这种起翻译作用的程序叫汇编程序，汇编程序是系统软件中的语言处理系统软件。汇编程序把汇编语言翻译成机器语言的过程称为汇编。机器语言和汇编语言属于低级语言，低级语言依赖于具体的机器类型，移植困难，学习较难，推广慢。

(3) 高级语言

高级语言比较多，现在可以在计算机上应用的编程语言有 100 多种，例如 Basic、Pascal、Fortran 等。高级语言有两种执行方式：解释方式和编译方式。解释方式如口译，每输入一句翻译执行一句，例如 Basic 语言有解释方式。编译方式必须将程序编译生成目标代码（后缀

名.obj),再连接生成可执行的机器语言(后缀名.exe)文件,才能执行。高级语言类似于数学语言,接近自然语言,容易编写、容易理解,不依赖于具体的计算机类型,可移植性强。有了高级语言,计算机的发展日新月异。

3. 程序设计

程序设计是给出解决特定问题程序的过程。程序设计往往以某种程序设计语言为工具,给出这种语言编写的程序。程序设计过程应当包括分析问题、确定算法、设计程序、输入编码、调试排错和测试等不同阶段。专业的程序设计人员常被称为程序员。

4. 程序设计的步骤

(1) 分析问题

对于接受的任务要进行认真的分析,研究所给定的条件,分析最后应达到的目标(输出的数据格式、输出何处、怎样输出等),找出解决问题的规律,选择解题的方法。

(2) 确定数据结构

根据任务的要求、指定输入数据和输出结果,确定存放数据的类型和数据的组织形式,即数据结构。

(3) 设计算法

设计出解题的方法和具体步骤,这就是设计算法。

(4) 编写程序

根据指定的数据结构和算法,用一种计算机语言编写程序,对源程序进行编辑、编译和连接。

(5) 运行程序,测试分析结果

运行可执行程序,得到运行结果。能得到运行结果并不意味着程序正确,要对结果进行分析,看它是否合理。若不合理要对程序进行调试,调试就是通过上机发现和排除程序中故障的过程。调试不仅要输入正确的数据,也要输入错误的数据和临界数据,不仅要达到有输出结果,还要输出结果的数据类型、有效数据位数、输出格式完全正确,完全符合问题的要求,才能算基本完成。

(6) 编写程序文档

绝大多数程序是提供给别人使用的,如同正式的产品应当提供产品说明书一样,正式提供给用户使用的程序,必须向用户提供程序说明书,内容应该包括:程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据以及使用注意事项等。

1.1.2 算法的概念和描述

1. 算法

简而言之,算法就是为解决某个特定问题而采取的准确而完整的方法与步骤。著名科学家沃思(Nikiklaus Wirth)曾提出一个公式:

$$\text{数据结构} + \text{算法} = \text{程序}$$

其中,数据结构是对数据的描述,也就是在程序中数据的类型和组织形式,而算法则是对操作步骤的描述。

算法是程序设计的灵魂,是问题求解过程中的精确描述,一个算法由有限条可以完全机械地执行的、有确定结果的指令组成。学习一门语言,不仅要熟练掌握语言本身的特点、语法规则等,更重要的是掌握分析问题、解决问题的方法,即锻炼分析、分解,最终归纳整理出算法的能力。解决一个具体问题时通常有多种算法供选用,要对算法执行效率进行分析。

2. 算法的特性

一个算法应该具有以下几个特性：

- 1) 有穷性：算法中的每个步骤执行的次数及时间是有限的。
- 2) 确定性：算法中的每条指令必须清楚严格地规定确切的含义，无二义性。
- 3) 有效性：算法中描述的每个步骤都应能有效地执行，并得到确定的结果。
- 4) 算法有零个或多个输入。
- 5) 算法有一个或多个输出。

3. 算法的描述方法

算法可以用各种描述方法来进行描述，历史上已经使用过的描述方法有自然语言、流程图、N-S 图、伪代码和计算机语言程序 5 种。

下面我们以计算 $1+2+3+\cdots+100$ 为例，使用不同的算法给予描述。

(1) 自然语言描述

步骤 1: $i=1$, $\text{sum}=0$;

步骤 2: 如果 i 不大于 100, 顺序执行步骤 3; 否则, 执行步骤 5;

步骤 3: sum 加上 i , 相加后的值仍放在 sum 中, 即 $\text{sum}=\text{sum}+i$;

步骤 4: 使 i 的值增 1 得到下一个加数, 即 $i=i+1$; 执行步骤 2;

步骤 5: 变量 sum 中的值就是所要得到的结果, 输出结果, 算法结束。

此方法口语化, 易懂, 但不够直观、不利于编程。

(2) 流程图描述

计算 $1+2+3+\cdots+100$ 的流程图如图 1-1 所示。传统流程图使用的图例如图 1-2 所示。

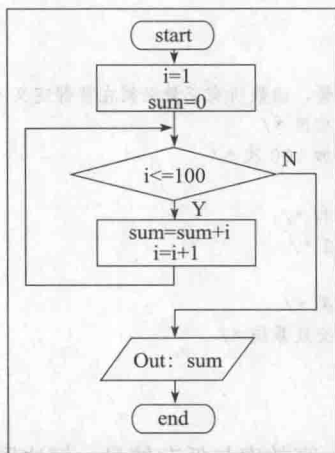


图 1-1 计算 $1+2+3+\cdots+100$ 的流程图

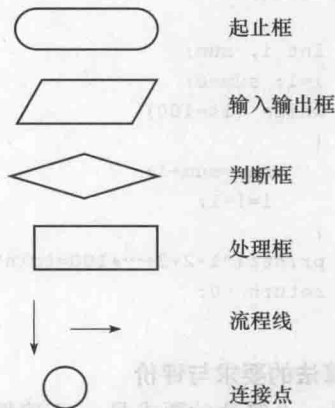


图 1-2 传统流程图使用的图例

用传统的流程图表示算法的优点是直观易懂、简单方便；缺点是传统流程图对于流程线的走向没有任何限制，可以任意转向，绘制流程图时费时费力且不易修改和阅读。注意带箭头的线段是流程线，它是控制流程的，所以又称为控制线。

(3) N-S 图描述

1973 年，美国学者 I.Nassi 和 B. Shneiderman 提出了另一种流程图描述方法，并以他们的名字命名为 N-S 结构化流程图。在这种流程图中，完全去掉了带箭头的流程线，全部算法写在一个矩形框内，在该框内还可以包含其他从属于它的框。计算 $1+2+3+\cdots+100$ 的 N-S 图

如图 1-3 所示。

用 N-S 图表示算法的优点是去掉了流程线，直观易懂，容易让学习者养成利用结构化方法构造算法的习惯，但修改流程图仍显麻烦。

(4) 伪代码描述（又称为类语言描述）

伪代码是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。它是人为的、非正式的语言，与日常用语类似，帮助程序员在写程序之前“设想出”程序，很容易被转换成程序代码，一般只包括可执行语句。这种方法比较易于理解，但描述较冗长。计算 $1+2+3+\cdots+100$ 的伪代码描述如图 1-4 所示。

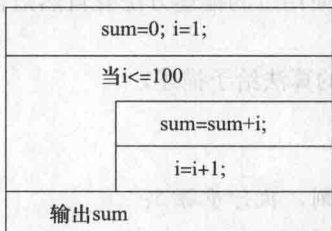


图 1-3 计算 $1+2+3+\cdots+100$ 的 N-S 图

```
i=1
sum=0
while i<=100
do
    sum=sum+i
    i=i+1
end while
print sum
```

图 1-4 计算 $1+2+3+\cdots+100$ 的伪代码描述

(5) 计算机语言程序描述

计算 $1+2+3+\cdots+100$ 的 C 语言程序描述。

【程序代码】 new1-0.c

```
/* 计算机 C 语言描述算法，文件名为 new1-0.c */
#include <stdio.h>
int main()
{
    int i, sum;
    i=1; sum=0;
    while (i<=100)
    {
        sum=sum+i;
        i=i+1;
    }
    printf("1+2+3+...+100=%d\n", sum);
    return 0;
}

/* 定义变量，函数所有变量必须在首部定义 */
/* 变量赋初值 */
/* 循环累加 100 次 */
/* sum 求和 */
/* i 递增 1 */
/* 输出结果 */
/* 控制权交还系统 */
```

4. 算法的要求与评价

设计一个算法的要求是：正确性、可读性、健壮性、高效率与低存储量。解决同一个问题的多个算法，执行时间越短，算法效率越高。对一个算法的评价主要是对算法效率的衡量，从时间复杂度和空间复杂度来衡量。

1) 时间复杂度：指执行算法所需要的时间。

2) 空间复杂度：指执行算法需要消耗的最大内存空间。

算法的复杂程度与运行该算法所需要的计算机资源的多少有关，所需要的资源越多，表明该算法的复杂性越高；所需要的资源越少，表明该算法的复杂性越低。对于同一个问题，根据不同的算法写出的程序放在计算机上运行时，所需要的时间和空间是不同的，两者都与问题的规模有关。