21世纪大学计算机教育系列规划教材

C/C++ 程序设计

李延珩 主编
张小兵 朱 斌 汪 青 李延珩 刘玉秀 矫桂秋 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

C/C++

21世纪大学计算机教育系列规划教材

C/C++程序设计

李延珩 主编

张小兵 朱 斌 汪 青
李延珩 刘玉秀 矫桂秋 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是根据教育部高等学校非计算机专业计算机基础课程教学指导分委员会《关于进一步加强高等学校计算机基础教学的意见》的教学基本要求并参考 2005 年版《全国计算机等级考试二级考试大纲（C 语言程序设计）》为高校非计算机专业学生编写的。

本书主要内容包括：C 语言基本概念和基础知识、结构化程序设计、数组、函数、文件、面向对象程序设计基础知识等。本书旨在将面向过程程序设计和面向对象程序设计有机地结合起来，使读者通过对本书的学习，能够具备开发小型应用系统的能力。

本书内容丰富、概念清晰、注重实践，适合作为高等院校各专业的程序设计课程教材，也可作为培训教材或自学教材。

图书在版编目（CIP）数据

C/C++程序设计/李延珩主编. —北京：中国铁道出版社，2007.9

（21 世纪大学计算机教育系列规划教材）

ISBN 978-7-113-08312-0

Ⅰ.C… Ⅱ.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 144797 号

书　　名：C/C++程序设计

作　　者：李延珩　等

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　秦绪好

责任编辑：崔晓静

特邀编辑：薛秋沛

封面制作：白　雪

责任校对：刘彦会

印　　刷：北京市彩桥印刷有限责任公司

开　　本：787×1092　1/16　**印张**：19.75　**字数**：461 千

版　　本：2007 年 9 月第 1 版　2007 年 9 月第 1 次印刷

印　　数：1～5 000 册

书　　号：ISBN 978-7-113-08312-0/TP・2582

定　　价：27.00 元

前言

FOREWORD

本书是根据教育部高等学校非计算机专业计算机基础课程教学指导分委员会《关于进一步加强高等学校计算机基础教学的意见》的教学基本要求并参考2005年版《全国计算机等级考试二级考试大纲（C语言程序设计）》为高校非计算机专业学生编写的。

随着计算机应用的普及与深入，对大学生的程序设计能力有了更高的要求。大学生既要掌握编写面向过程的结构化程序设计的方法，又要具备利用面向对象程序设计技术开发应用程序的能力。作者在多年的C语言教学中感到，目前的教学存在几个突出矛盾。一是“内容多”与“课时少”的矛盾，许多高校由于课时所限，不能单独开设“C语言程序设计”、“C++程序设计”和“Visual C++程序设计”3门课程，通常只能开设其中一门。二是对象与社会需求的矛盾。我们的授课对象通常是初学者，社会需要他们既要掌握面向过程程序设计的方法，又要具备面向对象程序设计的能力。实践证明对其直接开设面向对象程序设计技术课程，很多学生很难适应。若在Turbo C环境中讲授C语言，学生还需要自学C++。为此，作者对该门课程进行了改革和尝试，将传统的面向过程程序设计、现代的面向对象程序设计和MFC编程3个部分内容有机地结合成一门课程。在对传统的面向过程程序设计做较全面的介绍之后，又对面向对象程序设计和MFC编程有选择地进行介绍。希望学生通过该课程的学习，能直接利用Visual C++开发图形用户界面下的简单应用程序；在以后的学习或工作中，通过补充适当的知识就能够使用Visual C++编程解决各自专业领域的计算机应用问题。

本书共分12章，分为3个部分：前10章是传统的面向过程程序设计，内容包括基本概念和基础知识、结构化程序设计、数组、函数、文件等；第11章介绍面向对象程序设计基础知识，内容包括类和对象、继承和派生、多态性等；第12章介绍Visual C++编程基础知识，包括Windows界面编程和基于对话框的应用程序实现。

对于本书的学时安排，作者建议课堂教学30学时～54学时，上机实践30学时～54学时。为了克服内容多、课时少的矛盾，建议在课堂教学上强调培养学生的程序设计能力，语法问题让学生课后自学，对于复杂的问题通过案例教学化繁为简。

本书由李延珩主编。第1章、第2章、实验一和实验二由张小兵编写；第3章、第9章由朱斌编写；第4章、第6章、实验三、实验四、实验六和实验九由汪青编写；第5章、第10章、实验五、实验十和附录由李延珩编写；第7章、第8章、实验七和实验八由刘玉秀编写；第11章、第12章、实验十一和实验十二由矫桂秋编写。

在本书的编写过程中得到大连海事大学计算机科学与技术学院领导和许多老师的大力支持，同时得到了李湘一、陈策、陈书、周伟等研究生的帮助，在此一并表示感谢。

由于时间紧迫加之作者水平有限，书中难免存在疏漏之处，恳请广大读者批评指正。

编 者

2007年7月

CONTENTS

目录

第 1 章 C 语言程序设计概述 1
1.1 算法概念及描述 1
1.1.1 算法的概念 1
1.1.2 算法的描述 4
1.2 程序设计语言 7
1.3 C 语言的特点 8
1.4 C 语言的上机步骤 9
1.4.1 Turbo C 2.0 集成开发环境的安装和启动 10
1.4.2 Turbo C 2.0 集成开发环境的使用 10
1.4.3 编辑源文件 14
1.4.4 源程序的编译 15
1.4.5 程序的运行 15
1.4.6 程序的保存 15
1.5 简单的 C 语言程序举例 16
习题 18
实验一 初步了解 C 语言程序设计 19
第 2 章 数据类型、运算符和表达式 22
2.1 C 语言的数据类型 22
2.2 常量和变量 23
2.3 整型数据 25
2.3.1 整型数据说明 25
2.3.2 整型变量 26
2.3.3 整型常量 27
2.4 浮点数据（实型数据） 28
2.4.1 浮点数据说明 28
2.4.2 浮点变量 28
2.4.3 浮点常量 28
2.5 字符型数据 29
2.5.1 字符型数据说明 29
2.5.2 字符常量 29
2.5.3 字符变量 31
2.5.4 字符串常量 33
2.6 运算符及表达式 33
2.6.1 运算符及表达式概述 33
2.6.2 算术运算符和算术表达式 34

2.7 赋值运算符和赋值表达式 37
2.8 逗号运算符 39
2.9 不同数据类型间的转换 39
2.10 位运算 40
习题 44
实验二 数据类型、运算符和表达式 46
第 3 章 顺序结构程序设计 48
3.1 C 程序的基本结构及 C 语句的种类 48
3.1.1 结构化程序的 3 种基本结构 48
3.1.2 C 语句的种类 48
3.2 数据输入/输出的实现 49
3.3 标准输出函数——printf 函数 50
3.3.1 printf 函数的一般调用形式 50
3.3.2 printf 函数中常用的格式控制 51
3.3.3 调用 printf 函数时的注意事项 52
3.4 标准输入函数——scanf 函数 53
3.4.1 scanf 函数的一般调用形式 53
3.4.2 scanf 函数中常用的格式控制 54
3.4.3 调用 scanf 函数时的注意事项 55
3.5 字符输入/输出函数 56
3.5.1 字符输出函数 putchar 56
3.5.2 字符输入函数 getchar 56
3.6 顺序结构程序设计举例 57
习题 59
实验三 顺序结构程序设计 61
第 4 章 选择结构程序设计 63
4.1 关系运算符和关系表达式 63
4.1.1 关系运算符及其运算优先次序 63
4.1.2 关系表达式 64
4.2 逻辑运算符和逻辑表达式 64
4.2.1 逻辑运算符及其运算优先次序 64
4.2.2 逻辑表达式 65
4.3 if 语句 65
4.3.1 单分支 if 语句 65
4.3.2 双分支 if 语句 67
4.3.3 多分支 if 语句 68
4.3.4 if 语句的嵌套 70

4.4 条件运算符和条件表达式 72
4.5 switch 语句和 goto 语句 73
4.5.1 switch 语句 73
4.5.2 goto 语句 75
4.6 选择结构程序设计举例 75
习题 78
实验四 选择结构程序设计 83
第 5 章 循环结构程序设计 84
5.1 while 循环语句 84
5.2 do…while 循环语句 85
5.3 for 循环语句 87
5.4 循环的嵌套 88
5.5 循环的退出 90
5.5.1 break 语句 90
5.5.2 continue 语句 91
5.6 用 goto 语句构成循环 92
5.7 循环结构程序设计举例 93
习题 96
实验五 循环结构程序设计 100
第 6 章 数组 102
6.1 一维数组 102
6.1.1 一维数组的定义 102
6.1.2 一维数组的引用 103
6.1.3 一维数组的初始化 104
6.1.4 一维数组的应用 105
6.2 二维数组 111
6.2.1 二维数组的定义 111
6.2.2 二维数组的引用 111
6.2.3 二维数组的初始化 113
6.2.4 二维数组的应用 114
6.3 字符数组 115
6.3.1 字符数组的定义及结束标志 116
6.3.2 字符数组的初始化 116
6.3.3 字符数组的输入输出 117
6.3.4 字符串处理函数 119
习题 122
实验六 数组应用 127

第 7 章 函数 129
7.1 函数概述 129
7.2 函数的定义及说明（声明） 130
7.2.1 函数定义的格式 130
7.2.2 函数的说明（声明） 132
7.3 函数的调用 132
7.3.1 函数调用的格式 133
7.3.2 函数调用的方式 133
7.4 函数参数传递和函数的值 134
7.4.1 函数参数传递 134
7.4.2 函数的值 137
7.5 数组参数的传递 138
7.5.1 数组元素作为函数参数 139
7.5.2 数组名作为函数参数 140
7.6 函数的嵌套调用和递归调用 144
7.6.1 函数的嵌套调用 144
7.6.2 函数的递归调用 146
7.7 变量的存储类型与作用域 148
7.7.1 局部变量 148
7.7.2 全局变量 149
7.7.3 变量的存储类型 151
7.7.4 函数的作用域 155
7.8 编译预处理 156
7.8.1 文件包含 156
7.8.2 宏定义 158
7.8.3 条件编译 163
7.9 综合应用 164
习题 166
实验七 函数应用与预处理 170
第 8 章 指针 173
8.1 指针的概念 173
8.2 指针变量和指针运算符 175
8.2.1 指针变量 175
8.2.2 指针运算符 176
8.2.3 指针变量的运算 177
8.2.4 指针变量作为函数参数 183
8.3 指针与一维数组 186
8.3.1 引用数组元素的数组名法 186

8.3.2 指向数组元素的指针 187
8.3.3 数组名作为函数参数 189
8.4 指针与字符串 191
8.4.1 指向字符串的指针变量 191
8.4.2 字符串指针作为函数参数 194
8.5 指针与多维数组 196
8.5.1 多维数组的地址 196
8.5.2 多维数组元素的引用方法 198
8.6 指针数组与多级指针 201
8.6.1 指针数组 201
8.6.2 多级指针 203
8.6.3 指针数组作为 main 函数的形参 205
8.7 指针与函数 207
8.7.1 指向函数的指针 207
8.7.2 返回指针值的函数 209
习题 210
实验八 指针应用 213
第 9 章 结构体、共用体与枚举类型等构造数据类型 216
9.1 结构体 216
9.1.1 结构体说明 216
9.1.2 结构体变量的定义 217
9.1.3 结构体变量的引用 218
9.1.4 结构体变量的初始化 218
9.2 结构体数组 219
9.2.1 结构体数组的定义 219
9.2.2 结构体数组的引用 219
9.2.3 结构体数组的初始化 220
9.3 结构体与指针 220
9.3.1 结构体变量指针的定义 220
9.3.2 结构体数组指针 221
9.3.3 用结构体变量和指向结构体的指针作为函数参数 222
9.4 共用体 224
9.4.1 共用体类型的定义 224
9.4.2 共用体变量的定义 224
9.4.3 共用体变量的引用 224
9.5 枚举类型 225
9.6 用 typedef 定义类型 226
习题 228
实验九 结构体与共用体应用 232

第 10 章 文件 235
10.1 文件概述 235
10.1.1 文件的类型 235
10.1.2 文件类型指针 236
10.2 文件的打开与关闭 237
10.2.1 文件的打开 237
10.2.2 文件的关闭 237
10.3 文件的读写操作 238
10.3.1 读写一个字符的函数 238
10.3.2 读写一个数据块的函数 239
10.3.3 文件的格式化读写函数 241
10.3.4 读写一个字符串的函数 242
10.4 文件的定位 243
10.4.1 文件的顺序存取和随机存取 243
10.4.2 rewind 函数 243
10.4.3 fseek 函数 244
10.4.4 ftell 函数 244
10.5 程序设计举例 245
习题 247
实验十 文件操作 251
第 11 章 C++面向对象编程基础知识 253
11.1 面向对象的基本概念 253
11.2 类和对象的声明与访问 256
11.3 构造函数和析构函数 258
11.4 继承性与多态性 259
习题 262
实验十一 类和对象 263
第 12 章 Visual C++基础知识及应用 265
12.1 Visual C++开发环境简介 265
12.2 编写第一个 Visual C++项目 266
12.2.1 创建项目工作区 266
12.2.2 使用程序向导创建程序基本结构 266
12.2.3 设计窗体并添加 C++代码 267
12.3 Visual C++基本界面程序开发 270
12.4 综合应用 279
习题 287
实验十二 Visual C++练习 288
参考文献 291

附录 A 常用 ASCII 码表 292
附录 B C 语言中的关键字 293
附录 C 运算符和结合性 294
附录 D C 库函数 295
附录 E 《全国计算机等级考试二级考试大纲（C 语言程序设计）》 300

第 1 章 C 语言程序设计概述

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的，并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL（Basic Combind Programming Language）发展演变而来。在 1970 年，AT&T 贝尔实验室的 KenThompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言，最后推动了 C 语言的问世。

1.1 算法概念及描述

在计算机程序设计中计算机语言是人与计算机交流的工具，程序是指令的集合，而程序设计就是用计算机语言对所要解决的问题进行完整而准确的描述过程。一个完整的程序设计过程一般包含以下 4 个步骤。

（1）分析问题，建立数学模型。

（2）确定数据结构和算法。

（3）编写程序。

（4）调试程序。

数据结构是对程序中数据的描述；算法是对数据进行处理的步骤。在完成第一步后，要先确定数据结构和算法，然后用某种计算机语言来实现，即编写程序。而设计程序的目的就是要对数据进行处理以得到所期望的结果。编写程序实际上就是在实现某种算法。所以，算法在计算机程序设计中是非常重要的。

1.1.1 算法的概念

算法（Algorithm）是在有限步骤内求解某一问题所使用的一组定义明确的规则，即解决问题的方法和步骤。对计算机而言，就是计算机解题的过程与步骤。而具体语言，如 C 语言的语法是工具，是算法的一个具体实现。因此，在学习程序设计时，应熟练掌握语言的语法，因为它是算法实现的基础，同时也要认识到学习算法就是掌握分析问题、解决问题的方法，是锻炼分析、分解、最终归纳整理出算法的能力，是写出高质量的程序的关键之一。

下面通过例子来介绍如何设计一个算法。

【例 1.1】 输入 3 个数，输出其中最大的数。

首先，定义 3 个变量 a、b、c 分别存放 3 个数，再定义一个变量 max 存放最大的数。由于计算机一次只能比较两个数，首先比较 a 与 b，把大的数放入 max 中，再比较 max 与 c，把大的数放入 max 中。最后，输出 max。算法表示如下。

S1：分别输入 3 个数到 a、b、c。

S2：判断 a 是否大于 b。是，将 a 赋值给 max；否，将 b 赋值给 max。

S3：判断 c 是否大于 max。是，将 c 赋值给 max。

S4：输出 max 的值。

S5：结束。

【例 1.2】求 1+2+3+4+5+6 的结果。

先求出 1+2 的结果等于 3；再求出 3+3 的结果等于 6；再求出 6+4 的结果等于 10；再求出 10+5 的结果等于 15；再求出 15+6 的结果等于 21；最后，输出值 21。

考虑到每一次运算都是直接使用上一次的运算结果，上面算法可用循环算法来求结果。

首先，定义两个变量 i 和 s 来分别存放加数和计算结果。

S1：将 s 赋初值 0。

S2：将 i 赋初值 1。

S3：将 s+i 的结果赋值给 s。

S4：将 i+1 的结果赋值给 i。

S5：判断 i 是否大于 6。是，转到 S6；否，转到 S3。

S6：输出 s 的值。

S7：结束。

以上的算法已经可以很方便地转化为相应的程序语句。

由上例可看出，算法是一个有穷规则的集合，这些规则确定了解决某类问题的一个运算序列。对于该类问题的任何初始输入值，它都能一步一步地执行计算，经过有限步骤后终止计算并产生输出结果。归纳起来，算法具有以下基本特征。

（1）有穷性：一个算法必须在执行有限个操作步骤后终止。

（2）确定性：算法中每一步的含义必须是确切的，不可出现任何二义性。

（3）有效性：算法中的每一步操作都应该能有效执行，一个不可执行的操作是无效的。例如，一个数被 0 除的操作就是无效的，应当避免这种操作。

（4）有零个或多个输入：这里的输入是指在算法开始之前所需要的初始数据。这些输入的多少取决于特定的问题。例 1.1 的算法中有 3 个输入，即需要输入 a、b 和 c 三个初始数据。有些特殊算法也可以没有输入。

（5）有一个或多个输出：所谓输出是指与输入有某种特定关系的量，在一个完整的算法中至少会有一个输出。

任何简单或复杂的算法都是由基本功能操作和控制结构这两个要素组成的。计算机的基本功能操作包括以下 4 个。

（1）逻辑运算：与、或、非等。

（2）算术运算：加、减、乘、除等。

（3）数据比较：大于、小于、等于、不等于、大于等于、小于等于等。

（4）数据传送：输入、输出、赋值。

而控制结构是指各操作之间的执行顺序。

早期的非结构化语言中都有 goto 语句，它允许程序从一处直接跳转到另一处。这样做的好处是程序设计十分方便灵活，但其缺点也十分突出。一大堆跳转语句使得程序的流程十分

复杂紊乱，难以看懂，也难以验证程序的正确性，如果有错，找错十分困难。经过研究，人们发现，任何复杂的算法，都可以由顺序结构、选择（分支）结构和循环结构这三种基本结构组成，因此，我们构造一个算法的时候，也仅以这三种基本结构作为基础，遵守三种基本结构的规范，即基本结构之间可以并列、嵌套，但不允许交叉，不允许从一个结构直接转到另一个结构的内部去。正因为整个算法都是由三种基本结构组成的，就像用模块构建的一样，所以结构清晰，易于正确性验证，易于纠错，这种方法就是结构化方法。遵循这种方法的程序设计就是结构化程序设计。

1．顺序结构

顺序结构是简单的线性结构，各框按顺序执行，如图 1-1 所示。

2．选择（分支）结构

这种结构是对某个给定条件进行判断，条件为真或假时分别执行不同的框的内容，如图 1-2 所示。当条件为真时，执行语句块 1，否则执行语句块 2。

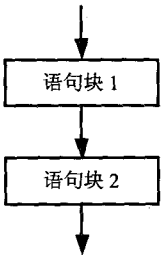

图 1-1　顺序结构

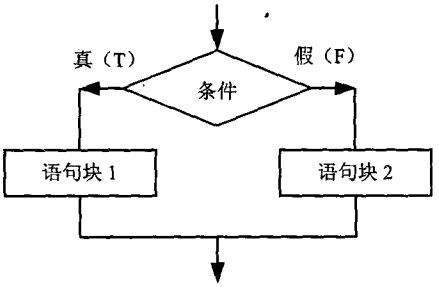

图 1-2　选择结构

3．循环结构

循环结构有两种基本形态：while 型循环和 do...while 型循环。

（1）while 型循环

如图 1-3（a）所示。

其执行序列为：当条件为真时，反复执行循环体，当条件为假，跳出循环，执行循环结构后面的语句。

（2）do...while 型循环

如图 1-3（b）所示。

执行序列为：首先执行循环体，再判断条件，条件为真时，再循环执行循环体，一旦条件为假，结束循环，执行循环结构后面的下一条语句。

上述三种基本结构都具有以下特点：

（1）只有一个入口和一个出口。

（2）结构内没有“死循环”，即无终止的循环。

（3）结构中的每一个块都有被执行的可能，即每个子结构都有一条从结构的入口到出口的路径通过。

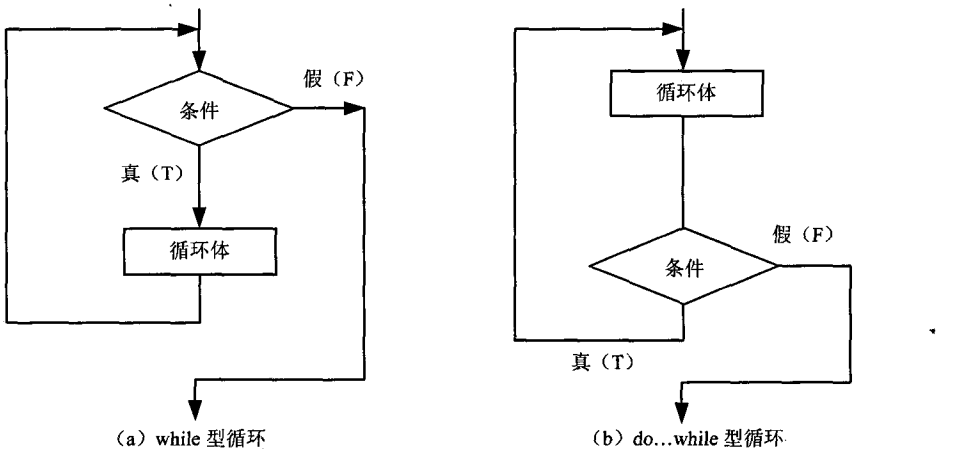

图 1-3 循环结构

1.1.2 算法的描述

算法可以用任何形式的语言和符号来描述，常用的有自然语言、流程图、N-S 图、PAD 图、伪代码等。所有的程序直接用程序设计语言表示算法。无论用哪种方法表示，都是为了清晰描述算法过程和解题步骤。

用自然语言表示算法，如例 1.1 所示，由于自然语言就是人们日常所用的语言，算法易懂，但文字太长，含义往往不严格。所以，除了很简单的问题外，一般不用。

流程图、N-S 图和 PAD 图是表示算法的图形工具，其中，流程图是最早提出的用图形表示算法的工具。它具有直观性强、便于阅读等特点，具有程序无法取代的作用。

N-S 图和 PAD 图符合结构化程序设计要求，是软件工程中强调使用的图形工具。

1. 流程图

流程图是一种传统的算法表示法，它利用几何图形的框来代表各种不同性质的操作，用流程线来指示算法的执行方向。因为它简单直观，易于理解，所以应用广泛，特别是在早期语言阶段，只有通过流程图才能简明地表述算法。直到结构化的程序设计语言出现，对流程图的依赖才有所降低。

图 1-4 所示是一些常用的流程图符号。

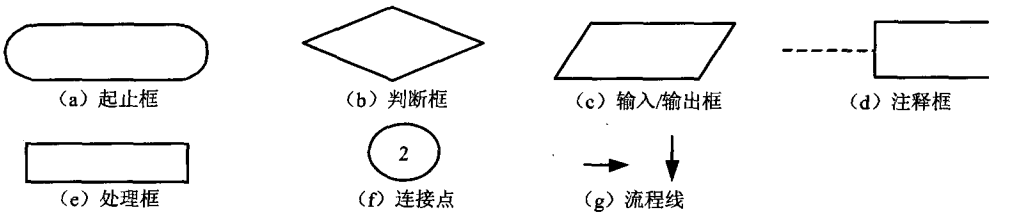

图 1-4 常用的流程图符号

起止框：表示算法的开始或结束。

判断框：表示根据条件决定程序的流向。

输入/输出框：表示数据的输入/输出操作。

处理框：表示初始化或运算赋值等操作。

连接点：用于将画在不同地方的流程线连接起来。圆圈内标号相同的点为同一点。使用连接点是为了避免流程线过长或交叉，以保证流程图的清晰。

图 1-5 和图 1-6 所示分别是用流程图表示例 1.1 和例 1.2 的算法。

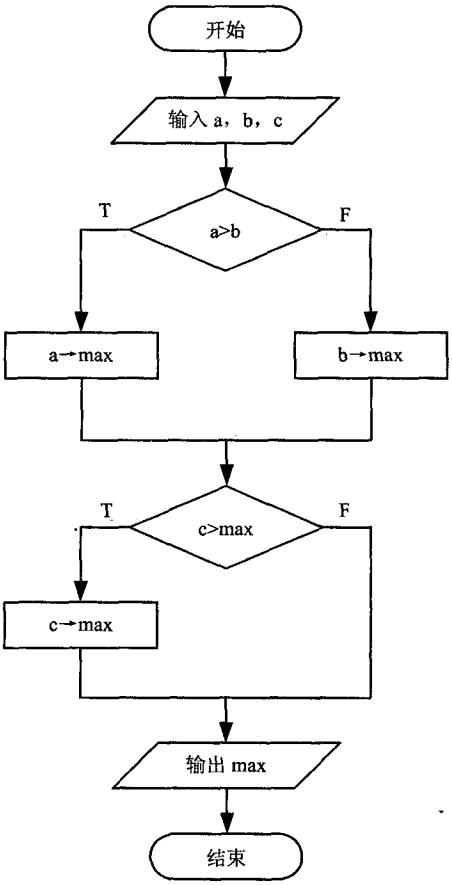

图 1-5　例 1.1 的算法流程图

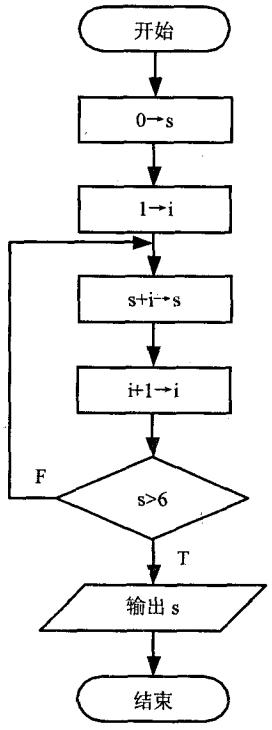

图 1-6　例 1.2 的算法流程图

2．N-S 图

N-S 图是另一种算法表示法，是由美国的 I.Nassi 和 B.Shneiderman 共同提出的，其根据是：既然任何算法都是由前面介绍的三种结构组成，所以各基本结构之间的流程线就是多余的，因此，N-S 图也是算法的一种结构化描述方法。

在 N-S 图中，一个算法就是一个大矩形框，框内又包含若干基本的框，三种基本结构的 N-S 图描述如图 1-7、图 1-8 和图 1-9 所示。

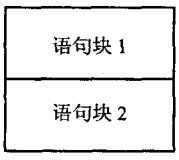

图 1-7　顺序结构的 N-S 图

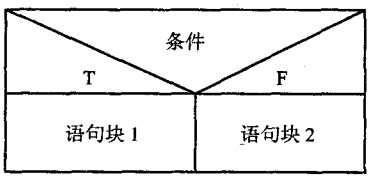

图 1-8　选择结构的 N-S 图