



高等教育精品课程规划教材

C语言程序设计

张红玲 畅惠明 主编



西南交通大学出版社

[Http://press.swjtu.edu.cn](http://press.swjtu.edu.cn)

高等教育精品课程规划教材

C 语言程序设计

主 编 张红玲 畅惠明
副主编 李辽辉 王亚子 王彦明
参 编 黄玉芳

西南交通大学出版社

· 成都 ·

图书在版编目(CIP)数据

C 语言程序设计 / 张红玲, 畅惠明主编. —成都:
西南交通大学出版社, 2014. 2
高等教育精品课程规划教材
ISBN 978-7-5643-2899-3

I. ①C… II. ①张… ②畅… III. ①C 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(2014)第 023656 号

高等教育精品课程规划教材

C 语言程序设计

主编 张红玲 畅惠明

责任编辑	张华敏
特邀编辑	唐建明 鲁世钊 陈正余
封面设计	水木时代(北京)图书中心
出版发行	西南交通大学出版社 (四川省成都市金牛区交大路 146 号)
发行部电话	028-87600564 028-87600533
邮政编码	610031
网 址	http://press.swjtu.edu.cn
印 刷	北京广达印刷有限公司
成品尺寸	185mm×260 mm
印 张	18.75
字 数	468 千字
版 次	2014 年 2 月第 1 版
印 次	2014 年 2 月第 1 次
书 号	ISBN 978-7-5643-2899-3
定 价	38.00 元

图书如有印装质量问题 本社负责退换

版权所有 盗版必究 举报电话:028-87600562

前 言

近年来,不少高校已将 C 语言作为学生学习程序设计的入门语言,教育部考试中心及部分省市将 C 语言程序设计纳入计算机等级考试的科目。C 语言以其小巧、灵活、高效等特点成为当今软件开发的主要编程语言。现在流行的 Visual C++ 和 C++ Builder 面向对象的程序设计,都是以 C 语言为基础的。

本书按照学生的认知规律和教学内容模块化的思想,以程序设计为主线,由浅入深地安排教学内容。本书共分八章,根据当前高校学生的实际情况,合理选取材料,精简概念,由易到难地介绍了 C 语言的基础知识及 C 语言程序设计方法,并通过实例进行说明。在本书的各章之后都提供了习题,供学生思考学习。

本书在编写过程中,为了更好地适应当前高校学生的特点,大量使用程序的运行结果解释 C 语言的知识,使用详细的分析引导读者完成 C 程序的设计。这样既可以减少教师备课的难度,又充分体现了以学生为主体的教育思想。不仅有利于教师教学,还有利于学生主动地自学。同时,考虑到多数学生都要参加全国计算机等级考试和计算机专业软件水平考试中的初级程序员考试,本书还介绍了和 Visual C++ 6.0 的使用方法和编译错误信息表及错误分析。

本书由张红玲、畅惠明任主编,李辽辉、王亚子、王彦明任副主编,黄玉芬任参编。具体分工如下:张红玲(辽宁科技学院)编写了第 5、6 章及 C 语言练习题与附录;畅惠明(辽宁科技学院)编写了第 2、3 章;李辽辉(辽宁科技学院)编写了第 4 章;王亚子(周口师范学院)编写了第 7 章;王彦明(辽宁科技学院)编写了第 1 章;黄玉芬(辽宁科技学院)编写了第 8 章。

本书是针对 C 语言初学者而编写的教材,内容安排合理、概念清晰、例题丰富、通俗易懂,是一本适合高校学生学习 C 语言的好教材,也可作为自学教材使用。

由于时间仓促和水平有限,书中错误和不当之处在所难免,敬请读者批评指正。

编 者

2014 年 1 月

目 录

第1章 软件开发简介	(1)
1.1 软件工程	(1)
1.2 程序介绍	(3)
1.3 数据结构	(10)
1.4 Visual C++6.0 集成开发环境	(12)
习题1	(19)
第2章 C语言的基础知识	(21)
2.1 C程序的基本结构	(21)
2.2 C语言的基本组成	(23)
2.3 数据类型概述	(25)
2.4 C语言的基本数据类型及其表示	(25)
2.5 常量与变量	(29)
2.6 运算符与表达式	(32)
2.7 位运算	(42)
2.8 混合运算及数据类型转换	(44)
2.9 数据的输入/输出	(45)
2.10 程序举例	(52)
习题2	(54)
第3章 C语言程序设计基本结构	(57)
3.1 结构化程序设计	(57)
3.2 选择结构程序设计	(59)
3.3 循环结构程序设计	(66)
3.4 程序举例	(72)
习题3	(77)
第4章 函 数	(82)
4.1 函数概述	(82)
4.2 函数的定义	(84)
4.3 函数的调用	(88)
4.4 函数的嵌套和递归调用	(92)
4.5 变量的存储属性	(97)
4.6 编译预处理	(106)
习题4	(110)

第 5 章 数 组	(114)
5.1 一维数组	(114)
5.2 二维数组	(120)
5.3 数组的应用	(125)
5.4 字符数组和字符串	(130)
5.5 数组作为函数的参数	(139)
5.6 程序举例	(143)
习题 5	(147)
第 6 章 指 针	(153)
6.1 指针与指针变量	(153)
6.2 指针与函数	(161)
6.3 指针与数组	(168)
6.4 指针与字符串	(176)
6.5 指针数组与命令行参数	(179)
6.6 程序举例	(184)
习题 6	(188)
第 7 章 结构体类型及其他构造类型	(192)
7.1 结构体类型	(192)
7.2 结构体数组	(197)
7.3 结构体指针	(201)
7.4 在函数之间传递结构体	(204)
7.5 用指针处理链表	(212)
7.6 共用体	(216)
7.7 用 typedef 定义类型	(219)
7.8 枚举类型	(221)
习题 7	(223)
第 8 章 文 件	(228)
8.1 文件概述	(228)
8.2 文件的打开与关闭	(229)
8.3 文件的读写	(232)
8.4 文件的定位	(241)
8.5 出错的检测	(243)
8.6 文件使用错误分析	(243)
习题 8	(244)
各章习题答案(部分)	(246)
C 语言练习题(一)	(248)
C 语言练习题(二)	(252)
C 语言练习题(三)	(257)

C 语言练习题(四)	(262)
C 语言练习题(五)	(267)
C 语言练习题(六)	(272)
C 语言练习题(七)	(277)
附录 I ASCII 字符编码表	(282)
附录 II C 语言中的运算符及其优先级	(283)
附录 III C 语言常用库函数	(285)
参考文献	(289)

第1章 软件开发简介

本章要点:本章对软件工程、数据结构、算法设计及程序设计的一些相关概念和基础知识作概括性介绍,通过本章的学习掌握软件工程的相关概念以及软件设计思想;理解并掌握常用的几种数据结构;掌握算法的概念、特征及常用的算法表示方法。

教学提示:本章的内容主要以理论为主,需要读者在理解的前提下掌握相关的知识要点。

1.1 软件工程

1.1.1 软件的概念及特点

1. 软件的定义

软件是程序、数据及其相关文档的完整集合。也可以说,软件由两部分组成:一是程序和文档;二是文档。文档是与软件的程序设计、开发、维护和使用有关的图文资料。

2. 软件的特点

①软件是一种逻辑产品,它与物质产品有很大的区别。它以程序和文档的形式出现,保存在计算机的存储器中(磁盘或光盘),通过计算机的运行才能体现它的功能和作用。

②软件产品的生产主要是研制。其成本主要体现在软件的开发和研制上,软件开发研制完成后,通过复制可以生产大量相同的软件产品。

③软件产品不会用坏,不存在磨损、消耗的问题,但是软件在使用过程中,往往需要根据用户需求的变化或软硬件环境的变化对软件进行修改,这种修改称为软件维护。

④软件产品的生产主要是脑力劳动。

⑤软件的费用是不断增加的,软件成本相当昂贵。

1.1.2 软件工程的相关知识

1. 软件工程

20世纪60年代中后期,软件数量越来越多,规模越来越大,而软件的生产基本上是各自为战,缺乏科学规范的系统规划与测试、评估标准,致使大批耗费巨资研制出来的软件系统因含有错误而无法使用,造成了巨大的损失。问题软件的不断出现,给人的感觉是软件越来越不可靠。这一切极大的震撼了计算机界,历史上称之为“软件危机”。一些软件开发人员认识到:大型程序的编制不同于小程序的编写,它应该是一项新的技术,应该像管理工程一样管理软件研制的全过程。于是,1968年北大西洋公约组织在联邦德国的一次学术会议上,首次提出了

“软件工程”的概念,从而形成了一门新兴的学科——软件工程。

软件工程是一门用工程化方法,指导计算机软件开发和维护的学科,它采用工程的概念、原理、技术和方法来开发、维护、管理软件。软件工程是目前计算机学科中一个年轻并且充满活力的研究领域。

2. 软件工程的 3 要素

软件工程包括 3 个重要要素,即方法、工具和过程。

①软件工程的方法是完成软件工程项目的手段。它包括许多方面,如项目计划和估算、软件需求分析、系统总体结构设计、数据结构设计、算法设计、编写代码、软件测试及维护等。

②软件工程的工具是人类在开发软件的活动中智力和体力的扩展和延伸,它提供了自动或半自动的软件支撑环境。

③软件工程的过程贯穿于软件开发的各个环节,支持软件开发各个环节的控制和管理。

3. 软件生命周期

任何有生命的动物或植物,都有一个生存周期,没有生命的事物或实体,如 PC(个人计算机)、家具、房子、汽车等,它们也有一个生存周期,这个生存周期就是使用寿命。软件和其他事物相似,有它的产生、发展和消亡的过程,软件生命周期是指从开发软件概念的提出起,直到该软件的使用、失去使用价值而被废弃的整个过程。

在软件生命周期中,每个阶段都有确定的任务,并产生一定规格的文档传递给下一阶段,下一阶段在前一阶段所提供的文档的基础上继续开展工作。软件生命周期包括 3 个时期 8 个阶段。软件定义期包括问题定义、可行性研究和需求分析 3 个阶段;软件开发期包括概要设计、详细设计、软件实现和软件测试 4 个阶段;运行维护期即运行维护阶段。

4. 软件开发模型

软件开发模型是软件开发全部过程、活动和任务的结构框架,它能直观地表达软件开发的全过程,明确规定要完成的主要活动、任务和开发策略。常用的软件开发模型有瀑布模型、快速原型模型、螺旋模型、基于构件的开发模型、基于知识的模型。

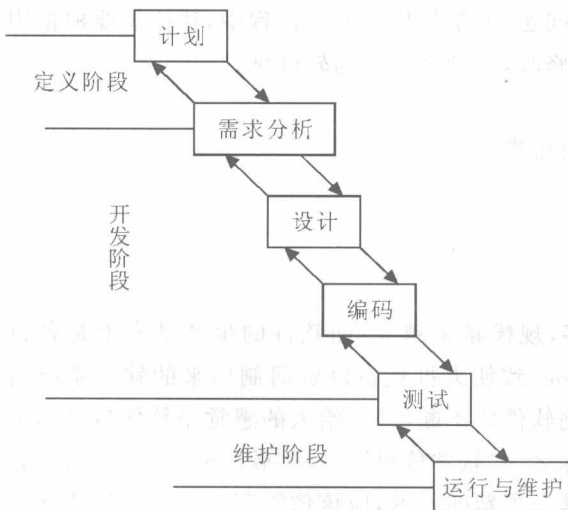


图 1-1 软件生存周期的瀑布模型

①瀑布模型严格按照生存周期各个阶段的目标、任务、文档和要求来进行软件开发,如图 1-1 所示。强调了每一阶段的严格性,尤其是开发前期的良好需求说明。开发人员试图在每一阶段活动结束后,通过严格的阶段性复审与确认,得到该阶段的一致、完整、准确和无二义性的文档,冻结这些文档作为该阶段的结束标志,并为下一阶段的活动奠定基础,从而形成一个理想的线性开发序列,以每一步的正确性和完整性来保证最终系统的质量。

瀑布模型在大量的软件开发实践中逐渐

暴露出了它的缺点。它是一种理想的线性开发模式,但缺乏灵活性,特别是无法解决软件需求不明确或不准确的问题。这些缺点给软件开发带来了严重的影响,最终可能导致开发的软件不是用户真正需要的软件,并且这一点在开发过程完成后才能发现。

②快速原型模型的第一步是快速建立一个能反映用户主要需求的原型系统,让用户在计算机上使用它,通过实践来了解目标系统的概貌。通常,用户试用原型系统之后会提出许多修改意见,开发人员按照用户的意见快速地修改原型系统,然后再次请用户试用。一旦用户认为这个原型系统确实能满足他们的需求,开发人员便可根据这份规格说明文档设计软件,此软件可以满足用户的真实需求。

5. 软件测试

软件测试是为了发现错误而执行程序的过程,其主要过程涵盖了整个软件生命期的过程。若从是否需要执行被测试软件的角度划分,软件测试方法和技术可以分为静态测试和动态测试方法;若按照功能可划分为黑盒测试和白盒测试。

①黑盒测试。黑盒测试完全不考虑程序内部的逻辑结构和内部特性,只依据程序的需求和功能规格说明,检查程序的功能是否符合它的功能说明。黑盒测试在软件的接口处。黑盒测试主要诊断功能错误或遗漏、界面错误,数据结构或外部数据库访问错误、性能错误、初始化和终止条件错误,主要用于软件的确认测试。

②白盒测试。白盒测试把测试对象看做一个打开的盒子,允许测试人员利用程序内部的逻辑结构及有关信息来设计或选择测试用例,对程序所有的逻辑路径进行测试。白盒测试在程序内部进行,主要用于完成软件内部操作的验证。

还有一些模型应用在软件工程里面,在本书中其他模型的开发过程可查阅软件工程方面的书籍。

1.1.3 软件工程的目标

软件工程是一门工程性学科,目的是成功地建造一个大型软件系统,同时达到这样几个目标:较低的成本、较好的性能、完善的功能、较好的移植性、较低的维护费用、及时的交付、较高的可靠性等。

1.2 程序介绍

1.2.1 程 序

当今社会,计算机已经广泛应用于社会生活的各个领域。但是,对于很多人来说,计算机还是很神秘的工具。事实上,计算机无非就是一种具有内部存储能力、由程序自动控制的电子设备。计算机本身只是一些简单硬件的组合,真正驱动其工作的是人们给它的指令,习惯上把这种命令计算机进行操作的一条条指令的集合称为“程序”。

计算机的工作由程序来控制,而程序的设计又是一项复杂的技术,需要相应的理论、技术、方法和工具来支持。就程序设计方法的发展而言,主要经过了结构化程序设计和面向对象的

程序设计阶段。

程序的编写需要用某种语言来完成,依据语言的发展,可以分为低级语言和高级语言,低级语言是直接接触机器硬件的机器语言(也是计算机可以直接识别的唯一语言)和汇编语言;而高级语言目前市面上有上百种,如 C 语言、Visual C++、Visual Basic、C++、Java 等,本书主要介绍 C 语言。

1.2.2 程序设计

1. 程序设计过程

程序设计就是用某种程序语言编写程序的过程。它是在多算法进行正确描述的基础上进行的,是用程序设计语言实现算法的过程。

编写程序的基本要求首先是保证语法的正确性,只有语法正确的程序才能通过编译系统的语法检查。然后是保证逻辑的正确性,这样程序执行后才能得到正确的结果。逻辑的正确性对于一个复杂的程序并不容易做到,需要一个反复测试和编辑的过程。一个简单的程序设计一般包括以下几个步骤:

①分析问题,建立数学模型。首先要对问题进行充分的分析,确定问题是什么,解决问题的步骤又是什么。将解题过程归纳为一系列的数学表达式,建立各种量之间的关系,即建立起解决问题的数学模型。

②确定数据结构。根据任务提出的要求,规划输入数据和输出结果,确定存放数据的数据结构。

③确定算法。针对所确定的数据结构确定解决问题的步骤。

④编制程序。根据确定的数据结构和算法,用程序设计语言编写程序,存入计算机中。

⑤调试程序。编译、调试原程序,修改语法错误与逻辑错误,直至程序运行成功。

⑥整理并写出文档资料。

2. 程序设计风格

除了好的程序设计方法和技术之外,程序设计风格也是很重要的。一般来讲,程序设计风格是指编写程序时所表现出的特点、习惯和逻辑思路。程序是由人来编写的,为了测试和维护程序,往往还要阅读和跟踪程序,因此程序设计的风格总体而言应该强调简单和清晰,程序必须是可理解的。良好的程序设计风格,主要注意以下几点:

(1) 源程序文档化

①符号的命名。符号名应具有一定的实际含义,即“见名知意”,以便理解程序的功能。

②程序注释。使用注释可以增强程序的可读性。注释一般分为序言性注释和功能性注释。

③视觉组织。在程序中利用空格、空行、缩进等技巧,采用锯齿形书写格式,使程序逻辑结构更加清楚,层次更加清晰。

(2) 数据说明

数据说明是指对程序中数据结构的组织和描述。数据说明一般应注意:数据说明的次序规范化;说明语句中变量安排有序化;使用注释说明复杂数据的结构。

(3) 语句的结构

语句是程序的基本单位,语句的构造力求简单、易懂、直接。编程时应保持“清晰第一、效率第二”的原则。

(4) 输入和输出

输入输出的方式和格式应尽可能方便用户的操作和使用。

1.2.3 程序设计的两大方法

1. 结构化程序设计

(1) 结构化程序设计方法的主要原则

结构化程序设计方法的主要原则可以概括为自顶向下、逐步求精、模块化。

①自顶向下。程序设计时,应先考虑总体,后考虑细节;先考虑全局目标,后考虑局部目标,先从最上层总目标开始设计,逐步使问题具体化。

②逐步求精。对复杂的问题,应设计一些子目标过渡,逐步细化。

③模块化。一个复杂问题,肯定是由若干个略微简单的问题构成。模块化是把程序要解决的问题分解成子问题;再进一步将子问题分解成子子问题,以此类推最后把每个小问题变为一个模块。

(2) 结构化程序设计的基本结构

采用结构化程序设计方法编写程序可使程序结构良好、易读、易理解、易维护。1966年,Boehm和Jacopini证明了程序设计语言仅仅使用顺序结构、选择(分支)结构、循环(重复)结构这三种基本控制结构就足以表达出各种其他形式结构的程序设计方法。

①顺序结构。顺序结构是三种基本结构中最简单的程序设计,也是最基本、最常用的结构,主要的执行过程按照语句的顺序、顺次执行即可。

②选择结构。选择结构又称为分支结构,它还可以细分为:单分支、双分支和多分支三种分支结构。它的执行主要依据程序中所给定的条件,依据条件来决定执行哪个分支语句。

③循环结构。循环结构又称为重复结构,其中还包括:for语句、while语句和do...while语句。它的执行主要是依据给定条件来决定是否执行循环体内的语句。

按照结构化程序设计方法设计程序的优点是:程序易于理解、使用和维护;提高了编程工作的效率,降低了软件开发的成本。

2. 面向对象程序设计

面向对象方法的主要优点是:与人类习惯的思维方式一致;稳定性好;可重用性好;易于开发大型软件产品;可维护性好。关于面向对象程序设计方法,涵盖了对象及对象的属性、方法、类、继承、多态性等几个基本要素。下面分别介绍面向对象程序设计方法中几个重要的基本概念。

(1) 对象

对象是面向对象程序设计方法中最基本的概念。对象可以用来表示客观世界中的任何实体,也就是说,应用领域中有意义的、与所要解决的问题有关系的任何事物都可以作为对象。例如,一个人、一家公司、一个窗口、一次通话、一场足球比赛等,都可以作为一个对象。总之,对象是对问题域中某个实体的抽象,设立某个对象就反应了软件系统保存有关它的信息并具

有与它进行交互的能力。

面向对象的程序设计方法中涉及的对象是系统中用来描述客观事物的一个实体,是构成系统的一个基本单位,它由一组表示其静态特征的属性和它可执行的一组操作组成。例如,一名学生是一个对象,它包含了学生的属性(学号、姓名、性别、系别、各门课程成绩等)及其操作(选课、考试、注册等),如对象 student 的属性有姓名、性别、学号等。不同对象的同一属性可以具有相同或不同的属性值,如张三的性别为男,学号 123456;李四的性别为男,学号 456789。

一般对象具有如下的特点:标识唯一性、分类性、多态性、封装性、模块独立性好。

(2)类和实例

将属性、操作相似的对象归为类。也就是说,类具有共同属性、共同方法的对象的集合。所以,类是对象的抽象,它描述了属于该对象类型的所有对象的性质,而一个对象则是其对应一个实例。

例如,student 是一个学生类,它描述了所有学生的性质。因此,任何一个学生都是学生类的对象,而一个具体的学生(如张三)是 student 类的一个实例。

(3)消息

消息是一个实例与另一个实例之间传递的信息,它请求对象执行某一处理或回答某一要求的信息,它统一了数据流和控制流。消息的使用类似于函数调用,消息中指定了某一个实例、一个操作名和一个参数表。接收消息的实例执行消息中指定的操作,并将形式参数与参数表中相应的值结合起来。消息传递过程中,由发送消息的对象的触发操作产生输出结果,作为消息传送到接收消息的对象,引发接受消息对象的一系列的操作。

例如,一个学生对象具有“选课”这项操作,那么要让学生上课的话需要学生根据学校要求和自己的能力有“选课”的消息。

一般消息由三部分组成:接受消息的对象的名称、消息标识符、零个或多个参数。

(4)继承

继承是面向对象程序设计方法的一个主要特征。继承是使用已有的类定义作为基础建立新类的定义技术。已有的类可当做基类来引用,则新类相应地可当做派生类来引用。广义上讲,继承是指能够直接获得已有的性质和特征,而不必重复定义它们。

在面向对象技术中,把类组成为具有层次结构的系统:一个类的上层可以有父类,下层可以有子类;一个类直接继承其父类的描述(数据和操作)或特征,子类自动地共享基类中定义的数据和方法。

继承的优点是相似的对象可以共享程序代码和数据结构,从而大大减少了程序中的冗余信息,提高软件的可重用性,便于软件修改与维护。

(5)多态性

对象根据所接受的消息而做出动作,同样的消息被不同的对象接受时可以导致完全不同的行动,该现象称为多态性。在面向对象的软件技术中,多态性是指子类对象可以像父类对象那样使用,同样的消息既可以发送给父类对象也可以发送给子类对象。

例如,在两个类 Male 和 Female 中都有一项属性为 Friend。一个人的朋友必须属于类 Male 或 Female,这是一个多态性的情况。利用多态性,用户能够发送一般形式的消息,而将所有的实现细节都留给接受消息的对象。

1.2.4 算 法

1. 算法的概念

算法是指为了解决某个特定问题而采取的确定的有限步骤和方法。程序设计过程中算法占有很大比重,一个算法的好与坏直接影响程序的质量。人们在日常的生产和生活中都离不开算法。例如,购物的“算法”是:确定需购买的商品、确定选购的商品、到达商场、购买商品、返回。

2. 算法的特征

一个好的算法应满足以下特性:

- ①有穷性。一个算法包含的操作步骤要在合理的时间内完成有限的操作,否则会出现“死循环”问题。
- ②确定性。算法中每一条指令都要具有明确的含义,不可以存在模棱两可的理解,也就是说确定的输入就必须有对应确定的输出。
- ③可行性。算法中的每一个步骤必须是通过验证可以实现的运算。
- ④输入可以有若干个。计算机处理的信息可以通过输入,也可以是计算机内部的信息。因此,计算机要处理的信息通过外部设备的输入可以包括零个,也可以包括多个。
- ⑤必须至少有一个输出。算法中,计算机处理完信息后必须有一个明确的结果。因此,计算机的信息输出至少要有一个。

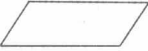
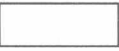
3. 算法的描述

由于自然语言等方式描述的算法存在着歧义性,因而在国际上确定了一些具有通用性的方法来描述算法,常用的算法描述有流程图、N-S 图、伪代码等。

(1)流程图

流程图是对给定算法的一种图形解法。它用规定的一系列图形、流程线及文字说明来表示算法中的基本操作和控制流程,其优点是形象直观、简单易懂、便于修改和交流。表 1-1 中分别列出了标准流程图的符号名称、符号和功能。

表 1-1 标准流程图符号

符号名称	符 号	功 能
起止框		表示算法的开始和结束
输入/输出框		表示算法的输入/输出操作,框内填写需输入或输出的各项
处理框		表示算法中的各种处理操作,框内填写处理说明或算式
判断框		表示算法中的条件判断操作,框内填写判断条件
注释框		表示算法中某操作的说明信息,框内填写文字说明
流程线		表示算法的执行方向
连接点		表示流程图的延续

例如,编写一个判断某年份是否是闰年的程序,其流程图见图 1-2。从流程图可以看出,该算法首先要求输入年份 n ,当 n 对 4 的余数不为 0 时,输出年份 n 不是闰年;否则,当 n 对 100 的余数不为 0 时,输出年份 n 是闰年;否则,当 n 对 400 的余数为 0 时,输出年份 n 是闰年;否则 n 年份非闰年。

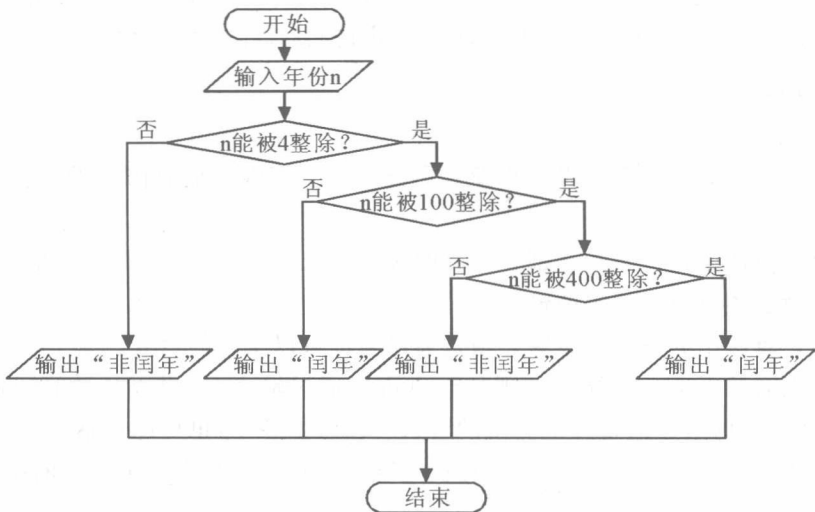


图 1-2 判断某年是否是闰年流程图

(2)N-S 图

虽然传统流程图可以描述算法中的绝大部分问题,但是流程图中流程线的使用没有严格限制,使得流程图变得毫无规律可言,给人们阅读和理解带来了一定的困难。为使流程图简洁、明晰,便于阅读,就必须限制滥用箭头,N-S 流程图很好地解决了这一问题。N-S 图的基本单元是矩形框。它只有一个入口和一个出口。长方形框内用不同形状的线来分割,可表示顺序结构、选择结构和循环结构,如图 1-3 所示。

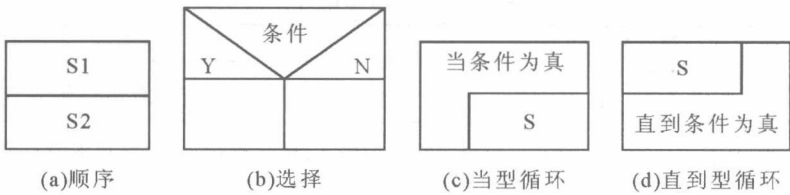


图 1-3 N-S 图

使用以上四种 N-S 流程图所使用的流程符号,可以组成复杂的流程图用于表示算法。图 1-4是用 N-S 图表示的判断某年份是否为闰年的流程图。

通过与传统流程图比较可以看出,用 N-S 流程图表示算法直观、形象,易于理解,比传统流程图紧凑易画,尤其是它废除了流程线,使得整个算法的结构由各个基本结构按顺序组成,其上下次序就是执行时的顺序,即位置靠上的先执行,位置靠下的后执行,写算法和看算法时只需从上到下进行就可以了,非常方便。

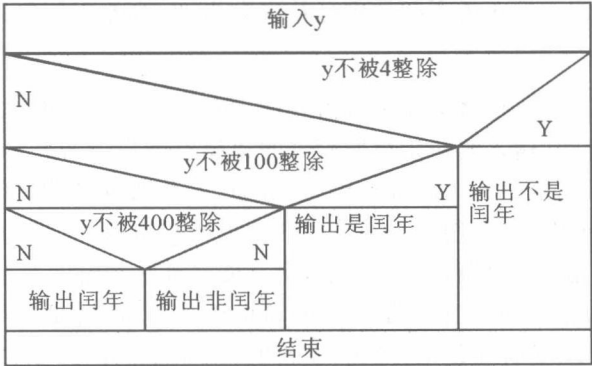


图 1-4 N-S 图判断某年是否是闰年

(3) 伪代码

伪代码是一种近似于高级语言但又不受语法约束的一种语言描述方式。其中，程序框架采用高级语言的关键字来表示，算法用接近于自然语言的形式来描述，如用伪代码来判断某年是否是闰年如下：

```
INPUT y
IF y MOD 4=0 THEN
  IF y MOD 100=0 THEN
    IF y MOD 400=0 THEN
      PRINTF“y 是闰年”
    ELSE
      PRINTF“y 非闰年”
    ENDIF
  ELSE
    PRINTF“y 是闰年”
  ENDIF
ELSE
  PRINTF“y 非闰年”
ENDIF
```

4. 算法的复杂度

算法的复杂度是算法效率的度量，是评价算法优劣的重要依据。一个算法的优劣体现在运行该算法所需要的计算机资源的多少上。计算机最重要的资源是时间和空间（存储器），所以，算法的复杂度主要包括时间复杂度和空间复杂度。

(1) 时间复杂度

所谓时间复杂度就是指执行算法所需要的计算工作量。算法的工作量用算法所执行的基本运算次数来度量，而算法所执行的基本运算次数是问题的规模，算法常用公式：

算法工作量 = f(n)

其中，n 是问题的规模。

例如,两个 n 阶矩阵相乘所需要的基本运算次数为 n^3 ,那么时间复杂度即为 n^3 。

(2)空间的复杂度

所谓空间复杂度,是指执行这个算法所需要的内存空间。一个算法所占的存储空间包括算法程序所占的存储空间、输入的初始数据所占的存储空间以及算法执行过程中所需要的额外空间。减低算法的空间复杂度,可从两个方面考虑:一是数据结构,即算法中所用到的所有数据,根据它们的实际范围,给它们定义适当的数据类型;二是算法,可用空间复杂度低的算法替代空间复杂度高的算法,这样做是有代价的,可能使得算法的时间复杂度提升了。

算法的空间复杂度和时间复杂度是一个有机的整体,有时降低时间复杂度(或空间复杂度)是以牺牲空间复杂度(或时间复杂度)为代价的。对于不同的问题,应具体问题具体分析,找出问题的最佳算法。

1.3 数据结构

1.3.1 数据的相关概念

①数据是对客观事物的符号表示,在计算机科学中是指能输入到计算机中并被计算机存储和加工的符号的总称。

②数据元素是数据的基本单位,可由若干个数据项组成,在程序中通常作为一个整体进行考虑和处理。

③数据项是数据不可分割的最小单位。

④数据对象是性质相同的数据元素的集合,是数据的一个子集,如工资管理项目中的职工即数据对象。

1.3.2 数据结构的概念及分类

数据结构是指相互之间存在一种或多种特定关系的数据元素的集合,是计算机存储和组织数据的方式。数据结构主要研究和讨论数据的逻辑结构、存储结构和数据的运算。

数据结构分为逻辑结构和存储结构两大类,具体分类如图 1-5 所示。

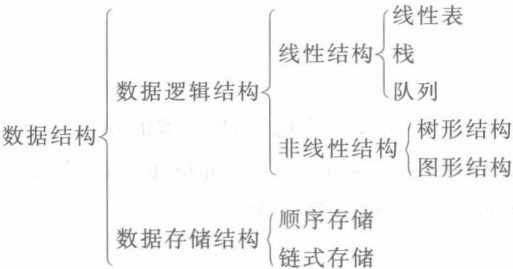


图 1-5 数据结构分类

1. 数据的逻辑结构

所谓数据的逻辑结构,是指反应数据元素之间逻辑关系的数据结构。数据元素之间有四