

全国计算机等级考试教程丛书

新大纲

全国计算机等级考试

三级教程

公共基础知识

姚普选 编著

- 紧扣最新的考试大纲
- 能力培养和应试并重
- 分解与综合合理安排
- 注重强化训练针对性



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

全国计算机等级考试教程丛书



全国计算机等级考试二级教程 ——公共基础知识

姚普选 编著

北京人民邮电出版社

北京

ISBN 7-113-02037-1

定价：18.00元

ISBN 7-113-02037-1

定价：18.00元

北京人民邮电出版社

ISBN 7-113-02037-1

定价：18.00元

ISBN 7-113-02037-1

定价：18.00元

ISBN 7-113-02037-1

定价：18.00元

中国铁道出版社

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是根据教育部考试中心制定的《全国计算机等级考试大纲（2004年版）》中对二级公共基础部分的要求而编写的。其中包括4方面的内容：数据结果与算法、程序设计基础、软件工程基础和数据库设计基础，每部分内容包括讲解、例题、习题三个部分。

本书在内容安排、讲解方式以及例题和习题的编排上都进行了精心独到的处理，不仅适合全国计算机等级考试（二级）的应考者使用；而且可以作为高等院校相应课程的教材或参考书。

图书在版编目（CIP）数据

全国计算机等级考试二级教程. 公共基础知识/
姚普选编著. —北京：中国铁道出版社，2006.8
（全国计算机等级考试教程丛书）
ISBN 7-113-07415-4

I. 全... II. 姚... III. 电子计算机—水
平考试—教材 IV. TP3

中国版本图书馆CIP数据核字（2006）第099868号

书 名：全国计算机等级考试二级教程——公共基础知识

作 者：姚普选

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：苏茜 赵轩 贾星

封面设计：薛为

封面制作：白雪

责任校对：张国成

印 刷：北京鑫正大印刷有限公司

开 本：787×1092 1/16 印张：11.25 字数：254千

版 本：2006年9月第1版 2006年9月第1次印刷

印 数：1~5 000册

书 号：ISBN 7-113-07415-4/TP·2037

定 价：16.00元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签，无标签者不得销售

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。

全国计算机等级考试教程

编委会名单

顾 问：刘瑞挺

主 编：冯博琴

副主编：艾德才 徐建民

编 委：(按姓氏字母的先后为序)

边莫英	高寅生	贺亚茹	蒋金丹	李 煦
李亚平	李尊朝	林成春	刘振鹏	路景贵
卢素魁	宁书林	邵 静	宋汉珍	宋志清
王铁凡	王祖卫	吴建平	张 鹏	左建中

序

全国计算机等级考试是面向全社会的计算机应用能力水平考试，自 1994 年举办以来，该项考试持续发展，享有良好的社会信誉，已成为我国普及计算机教育不可或缺的组成部分，是国内参加人数最多、影响最大的计算机水平考试。

全国计算机等级考试是经教育部批准，由教育部考试中心主办，测试应试者计算机应用知识与能力的等级水平考试。其目的在于以考促学，向社会推广和普及计算机知识，也为用人单位录用和考核工作人员提供一个统一、客观、公正的标准。由于计算机等级考试作为教育部推出的一种社会考试，具有权威性、科学性和公平性，反映了一个人在某一方面的能力，其证书由于符合人才市场的需要而受到各方面的欢迎，而证书的取得需要广大考生调整好心态，既不要畏首畏尾，又不要心存侥幸。希望在这套丛书的帮助下，扎扎实实地学习知识，在理解的基础上掌握，这样才能取得好的成绩，获得国家认证的合格证书。

本丛书是根据教育部考试中心最新考试大纲，并在研究近年来全国计算机等级考试教程的基础上编写、出版的。

该套丛书特点：创意新颖，实用性强；结构清晰，讲解透彻。具体体现在以下四点。

● 体现新大纲要求

本丛书根据教育部考试中心制定的最新考试大纲要求编写。每本书覆盖了该门课程在大纲中所提到的所有内容。全书的结构清晰，学习目标非常明确。

● 体现能力培养和应试并重的一致性

读者在学习过程中学的不仅是一种知识，更是一种技能。本丛书在写作上主要从应用的角度构建知识体系，以培养能运用所学知识的人才为目标，因此技能训练是非常必要的。本丛书在具体内容安排时，以突出其实用性为目标，将复杂的理论融于具体的实例之中。每本书中的大量实例都是经过精心设计的，通过各个实例把一些知识点有机地串联起来，从而使本丛书的逻辑性更强，让读者接受得更快。

但是等级考试又是一种考试，不仅仅考运用的能力，也考有关计算机操作和语言等的基本知识，如何科学安排这样的交叉是个很关键的问题，本丛书在这方面安排得非常好，相应的教材也在写作上注意到了有关知识点的讲解。每本书在每章的最后都安排了“典型例题解析”，注重解题思路，传授解题方法，对读者参加计算机等级考试提供了很大帮助。

● 注重分解与综合

人们认识复杂事物的一个基本方法是分解与综合，对学生来说，一门新课相当于是一个需要接受的新事物，采用“分解与综合”的方法可以使他们感觉到学习更容易。因此本教材使内容尽量大小适中，即每一章、每一节乃至一个知识点都尽可能保持适中，将难点适当分解，便于学生掌握。

● 注重强化训练

为了让读者扎实地掌握所学知识，并且在等级考试中考出好成绩，本丛书在每章的后面都安排有一定量的习题。在习题安排上基本都与考试的类型一致，并且提供了习题答案。除此之外，每本书在最后都附有两套模拟题，供读者进行自我水平的检测，也可作为读者考前冲刺用。并且，在丛书的一些书中安排了编译环境与上机考试流程的内容，这样让读者对考试环境有所了解，进而做到心中有数。

本丛书的作者都是高等院校或计算机应用部门中具有丰富教学经验并对计算机等级考试有深入研究的教授、专家。

中国铁道出版社计算机图书中心一直以“热忱、专业、卓越品质”为信念出版各类计算机技能培训图书、教材和专业辅导书。我们希望所出版的每一本图书都能对读者的学习和工作带来帮助。如果在使用本书中碰到一些问题，可以到 <http://www.tqbooks.net> 上提问，我们的编辑和作者将热忱为读者服务。

本丛书如有不足之处，敬请各位专家、老师和广大读者不吝指正。我们相信，此套丛书的出版，将对读者的计算机学习起到帮助作用，也必将对我国的计算机等级考试的发展起到推进作用。

丛书编写委员会

2005年9月

前 言

本书是根据教育部考试中心制定的《全国计算机等级考试大纲（2004 年版）》中对二级公共基础部分的要求而编写的。

全书共分 4 章。

第 1 章的主要内容包括：算法的概念及算法复杂度分析的基本概念，数据结构的定义和基本概念，线性表的概念及基本运算，栈和队列的概念及基本运算，线性链表的概念及基本运算，二叉树的概念、存储结构及其遍历，以及几种常用的查找和排序算法。

第 2 章的主要内容包括：程序设计的方法与风格，结构化程序设计方法、面向对象程序设计方法的基本概念。

第 3 章的主要内容包括：软件工程的基本概念，结构化分析方法，结构化设计方法，软件测试的基本方法，以及程序的调试方法。

第 4 章的主要内容包括：数据库、数据库管理系统和数据库系统的概念，E-R 数据模型和 E-R 图，数据模型的概念，关系代数中的基本运算，以及数据库设计的基本方法和步骤。

每章中还包括了与实际考试的题型相同、难度相当的例题和习题两部分内容。其中，例题部分给出了答案及部分题目的详细分析。习题答案统一放在本书最后。

考虑到公共基础知识虽然是每套试卷中必有的内容，但毕竟只是其中的一小部分，应试者一般不太可能或不太情愿花费太多的精力来学习这部分内容，因此，笔者根据各部分内容之间的内在联系以及学习活动中的情况，对于重要的名词、抽象的概念、常用的技术，以及具有递进关系的系列内容的讲解，都进行了精心的安排。例如，所涉及的名词、概念和方法尽量给出明确的结论及详细的讲解，读者一般不再需要翻阅其他书籍就可以理解。又如，书（正文）中的例题在保证其典型性、正确性的前提下，尽量采用能够套用的内容和形式，以达到巩固甚至加深理解的作用。

由于时间仓促及编者水平有限，书中可能存在不妥甚至错误之处，恳请读者批评指正。

编 者

2006 年 6 月于西安交通大学

目 录

第 1 章 数据结构与算法	1
1.1 算法的概念	1
1.1.1 算法实例	2
1.1.2 算法的特征	4
1.1.3 算法的基本要素	5
1.2 算法复杂度	9
1.3 数据结构的概念	12
1.3.1 数据结构研究的对象	12
1.3.2 数据结构与算法	14
1.4 线性表	15
1.4.1 线性表的概念	16
1.4.2 线性表的顺序存储结构	16
1.4.3 线性表的链式存储结构	19
1.5 受限线性表	22
1.5.1 栈	23
1.5.2 队列	24
1.6 树与二叉树	27
1.6.1 树的概念	27
1.6.2 二叉树的概念及性质	28
1.6.3 二叉树的存储结构	30
1.6.4 二叉树的遍历	31
1.7 查找	33
1.7.1 查找的概念	33
1.7.2 常用查找方法	34
1.8 排序	35
1.8.1 排序的概念	36
1.8.2 插入排序	37
1.8.3 交换排序	39
1.8.4 选择排序	41
1.9 例题	43
习题 1	49
第 2 章 程序设计基础	52
2.1 程序设计方法与风格	52
2.2 结构化程序设计	53
2.2.1 结构化程序设计的产生	54
2.2.2 结构化程序设计原则	54

2.2.3	程序的三种基本结构	55
2.2.4	基本结构的本质属性	58
2.2.5	结构化程序的特点	59
2.2.6	结构化程序设计原则和方法的应用	60
2.3	面向对象程序设计	61
2.3.1	面向对象程序设计思想	61
2.3.2	面向对象程序设计的优点	62
2.3.3	对象和类	65
2.3.4	消息	66
2.3.5	类的继承性	67
2.3.6	类的多态性	69
2.4	例题	69
习题 2	72
第 3 章	软件工程基础	73
3.1	软件工程基本概念	73
3.1.1	软件定义与软件特点	73
3.1.2	软件危机与软件工程	74
3.1.3	软件工程过程	76
3.1.4	软件生命周期	77
3.1.5	软件工程的目标与原则	78
3.1.6	软件开发工具与环境	80
3.2	结构化分析方法	80
3.2.1	需求分析与需求分析方法	81
3.2.2	结构化分析方法	82
3.2.3	软件需求规格说明书	86
3.3	结构化设计方法	88
3.3.1	软件设计的基本概念	88
3.3.2	软件设计的基本原理	89
3.3.3	概要设计的任务	91
3.3.4	面向数据流的设计方法	94
3.3.5	详细设计	97
3.4	软件测试	101
3.4.1	软件测试的目的、准则与方法	101
3.4.2	白盒测试及测试用例设计	103
3.4.3	黑盒测试及测试用例设计	106
3.4.4	软件测试的实施	109
3.5	程序调试	113
3.5.1	基本概念	113
3.5.2	软件调试方法	114

3.6 例题.....	115
习题 3.....	120
第 4 章 数据库设计基础.....	122
4.1 数据库系统基本概念.....	122
4.1.1 数据管理技术的发展.....	122
4.1.2 数据、数据库与数据库管理系统.....	125
4.1.3 数据库系统及其特点.....	127
4.1.4 数据库系统内部体系结构.....	129
4.1.5 数据库技术的现状及发展趋势.....	132
4.2 数据模型.....	134
4.2.1 数据模型的概念.....	134
4.2.2 E-R 模型.....	135
4.2.3 层次模型.....	138
4.2.4 网状模型.....	140
4.2.5 关系模型.....	140
4.3 关系代数.....	144
4.3.1 关系的概念.....	144
4.3.2 关系的基本运算.....	145
4.3.3 关系代数中的扩充运算.....	147
4.3.4 关系代数应用实例.....	150
4.4 数据库设计与管理.....	151
4.4.1 数据库设计概述.....	151
4.4.2 数据库设计的需求分析.....	152
4.4.3 数据库概念设计.....	153
4.4.4 数据库的逻辑设计.....	156
4.4.5 数据库的物理设计.....	159
4.4.6 数据库管理.....	159
4.5 例题.....	160
习题 4.....	164
习题答案.....	167

第 1 章 数据结构与算法

算法是计算机科学的重要基础，而数据结构又是算法研究的基础，因而，数据结构是计算机科学研究的基本课题。全国计算机等级考试二级的公共基础部分涉及到的数据结构和算法的内容包括以下几个方面：

- (1) 算法的基本概念；算法复杂度的概念和意义（时间复杂度与空间复杂度）。
- (2) 数据结构的定义；数据的逻辑结构与存储结构；数据结构的图形表示；线性结构与非线性结构的概念。
- (3) 线性表的定义；线性表的顺序存储结构及其插入与删除运算。
- (4) 栈和队列的定义；栈和队列的顺序存储结构及其基本运算。
- (5) 线性单链表、双向链表与循环链表的结构及其基本运算。
- (6) 树的基本概念；二叉树的定义及其存储结构；二叉树的前序、中序和后序遍历。
- (7) 顺序查找与二分法查找算法；基本排序算法（交换排序、选择排序、插入排序）。

数据是由计算机进行加工处理的描述客观事物的信息。由于客观事物的多样性，由描述客观事物的属性抽象而成的数据就有各种不同的形式，如整数、实数、字符、图像、动画、声音、视频等。为了描述客观事物之间的联系，数据之间的联系又被抽象为各种不同的数据结构。数据结构既指数据对象及其相互关系和构造方法，又指研究数据的组织、构造、性质、规律等的计算机科学分支，数据结构是计算机科学研究的基本课题。

计算机之所以能够处理各种不同的问题，是因为人们编制了处理问题的程序，由计算机来执行程序而得到结果。然而，程序总是依据某种解决问题的算法编制出来的。解决同一个问题往往可以使用几种算法，根据不同的算法编制出来的程序自然会有高下优劣之分，因此，在学习程序设计方法时，既要学习程序设计语言或软件开发环境的使用规则，还要学习如何根据实际问题来设计正确高效的算法。可以认为，算法设计是程序设计的关键所在。

算法与数据结构是人们用计算机解题时所进行的两种抽象，算法是从计算机操作的角度对解题过程的抽象，数据结构是从如何组织被处理对象的角度进行抽象。这两种抽象互相依赖、互相补充，可以有效地降低问题的复杂性。例如，将一批具有顺序关系的数据放在一个数组或文件中，就可以使用循环结构来处理，比只使用简单变量的效率高得多。

理论和实践都证明，不了解施加于数据之上的算法，就不知道如何组织和构造数据；反之，算法的结构和选择，常常在很大程度上依赖于数据的组织与构造。

1.1 算法的概念

算法是程序设计的根据，程序是算法的具体实现，算法和程序自然有许多相同的地方，但算法不等于程序，也不等于计算方法。算法是解题方案的准确而完整的描述，是由一系列求解特定问题的“语句”或“指令”构成的，而程序受计算机运行环境的限制，需要考虑许多与方法和分析无关的细节问题。一般来说，实现某个算法的程序不会优于这个算法。

【注】“指令”是机器语言或汇编语言中的名词，一条指令就是一个描述动作的语句。

1.1.1 算法实例

设计算法的目的是为了解决实际问题，因此，一个算法就是解决某种实际问题的一套操作规则。按照人们解决问题的一般方法，一个算法应该能够在一定的时间内，根据现有条件（如有限的空间），逐步地对给定的原材料进行适当的加工处理，然后得到预期的结果。

【例 1-1】给一个简单的计算器程序编写算法。

计算器程序的任务是：输入两个数，指出运算的类别，然后对两个数进行某种运算，最后输出结果。据此可编写如下算法。

S1: 输入两个操作数 m 和 n 。

S2: 输入“运算类别”。

S3: “正常” \Rightarrow MessageError。

S4: 判断运算类别是什么。

S4.1: 若为加法，则 $m+n \Rightarrow$ Result。

S4.2: 若为减法，则 $m-n \Rightarrow$ Result。

S4.3: 若为乘法，则 $m \times n \Rightarrow$ Result。

S4.4: 若为除法，则判断 $n \neq 0$? 是则执行 S4.4.1，否则执行 S4.4.2。

S4.4.1: $m \div n \Rightarrow$ Result。

S4.4.2: “0 不能做除数!” \Rightarrow MessageError。

S5: 判断 MessageError=“正常”? 是则执行 S5.1，否则执行 S5.2。

S5.1: 输出 Result。

S5.2: 输出 MessageError。

S6: 算法结束。

【例 1-2】求两个自然数 m 和 n 的最大公约数（能同时整除 m 和 n 的最大正整数）。

我国古代数学名著^{注1}《九章算术》方田章中为了约简分数 $\frac{m}{n}$ ，提出了求“等数”（最大公约数）的更相减损法（今称欧几里得算法或辗转相除法），即“以少减多，更相减损，求其等也”。按照这种思想编写的算法如下。

S1: 输入两个自然数 m 和 n 。

S2: 判断 $m < n$?

是， m 和 n 互换。

S3: 反复执行以下操作，直到 $r=0$ 。

S3.1: $m \div n$ ，余数 $\Rightarrow r$ 。

S3.2: 判断 $r \neq 0$?

是， $n \Rightarrow m$ ， $r \Rightarrow n$ 。

S4: 输出 n 。

S5: 算法结束。

注 1 见《九章算术校证》，李继闵著，陕西科学技术出版社，1993 年 9 月

一般来说, 一个算法都有一定的抽象性。可以通过若干典型实例, 验证算法对它所涉及的各种情形的正确性。例如, 更相减损法可通过以下实例来验证。

1. $m > n$ 时的验证

当 $m > n$ 时, 不妨令 $m=102$, $n=54$, 则算法的执行过程如下。

S1: $102 \Rightarrow m$, $54 \Rightarrow n$ 。

S2: 因 $102 > 54$, 故不必交换。

S3: 循环: 第 1 次

S3.1: $102 \div 54 = 1$ 余 48 ($\Rightarrow r$)。

S3.2: 因 $48 \neq 0$, 故 $54 \Rightarrow m$, $48 \Rightarrow n$ 。

S3: 循环: 第 2 次

S3.1: $54 \div 48 = 1$ 余 6 ($\Rightarrow r$)。

S3.2: 因 $6 \neq 0$, 故 $48 \Rightarrow m$, $6 \Rightarrow n$ 。

S3: 循环: 第 3 次

S3.1: $48 \div 6 = 8$ 余 0 ($\Rightarrow r$)。

S3.2: 因 $0=0$, 故跳出循环。

S4: 输出 6 (n 的值)。

S5: 结束。

2. $m = n$ 时的验证

当 $m = n$ 时, 不妨令 $m = n = 98$, 则算法的执行过程如下。

S1: $98 \Rightarrow m$, $98 \Rightarrow n$ 。

S2: 因 m 和 n 相等, 故不必交换。

S3: 循环: 第 1 次

S3.1: $98 \div 98 = 1$ 余 0 ($\Rightarrow r$)。

S3.2: 因 $0=0$, 故跳出循环。

S4: 输出 98 (n 的值)。

S5: 结束。

3. $m < n$ 时的验证

当 $m < n$ 时, 不妨令 $m=54$, $n=102$, 则算法的执行过程如下。

S1: $54 \Rightarrow m$, $102 \Rightarrow n$ 。

S2: 因 $m < n$, 故 m 和 n 互换。

.....

(以下操作类同于 $m=102$, $n=54$ 时的操作, 故略。)

严格地说, 只有经过证明的算法才是正确可靠的算法, 而以实例来验证算法未必就是算法的证明, 一般地, 证明比验证要困难得多。例如, 在欧几里得《几何原本》中, 证明了更相减损法的前两步:

如果 $m > n$, $m \div n$ 的余数 $r_1 = 0$, 则 n 是 m 和 n 的最大公约数。

如果 $m \div n$ 的余数 $r_1 \neq 0$, 而 $n \div r_1$ 的余数 $r_2 = 0$, 则 r_1 是 m 和 n 的最大公约数。

而在《九章算术》中, 刘徽注给出了这种算法的依据: “其所以相减者, 皆等数之重叠,

故以等数约之”。即 m 和 n 以及最后一次除法之前的各次余数都是最大公约数的倍数。

对于同一个问题，往往可以写出多种不同的算法。例如，对于求两个自然数的最大公约数这个问题，可以这样考虑：用两个数中较小的一个作为测试数，试除 m 和 n 。若能同时整除，则测试数即为最大公约数；不能除尽时，测试数减去 1，再试除，……如此反复，直到测试数能同时整除 m 和 n 为止，此时的测试数即为最大公约数。据此思路编制的算法如下。

S1: 输入两个自然数 m 和 n 。

S2: 判断 $m > n$? 是则执行 S2.1, 否则执行 S2.2。

S2.1: $n \Rightarrow i$ 。

S2.2: $m \Rightarrow i$ 。

S3: 当 $m \div i$ 余数 $\neq 0$ 或 $n \div i$ 余数 $\neq 0$ 时，反复执行以下操作：

$i-1 \Rightarrow i$ 。

S4: 输出 i 。

S5: 算法结束。

将这个算法与更相减损法比较，当两个自然数 m 和 n 较大，且最大公约数较小时，这个算法需要执行的循环次数比前面的算法多得多，因此，这是一种效率比较低的算法。但考虑到计算机的特点，用这个算法来编制程序同样是安全可靠的。另外，由于这个算法的思路要简单得多，所以，也是经常采用的一种算法。

1.1.2 算法的特征

算法是程序设计的根据，程序设计的实质就是构造解决问题的算法，并将其解释为程序设计语言。因而，算法设计的质量将直接影响程序设计的质量。

算法的应用无所不在，但却很难给出严格的定义，一般来说，计算机科学中的算法是由一套规则组成的一个过程。所谓过程就是一些步骤，这些步骤集中在一起能给出一类问题的解答。因此，算法实际上是一种抽象的解决问题的方案。通常认为一个算法必须具备有穷性、确定性、数据输入、输出和可行性这 5 个基本特征。

1. 有穷性

有始有终是算法最基本的特征。换言之，一个算法必须在它所涉及的每一种情况下，都能在执行有穷步操作之后结束。例如，更相减损法就具备这一特征，无论两个自然数 m 和 n 是多少，都会在循环若干次之后，求得它们的最大公约数。 m 和 n 越大，需要循环的次数就会越多，但只不过是增加了有限次循环而已。

有穷性实际上还包含了可以容忍的合理限度，如果一个算法要执行几千年才能结束，实际上也是没有意义的。对这类问题的求解应另辟蹊径。

应该指出，程序是使用某种程序设计语言的算法的具体实现。程序可以不满足有穷性这一要求。例如，操作系统就是一个在无限循环中执行的程序，因而不是一个算法。然而可以把操作系统的各种任务看成是多个单独的问题，每一个问题由操作系统中的一个子程序通过特定的算法来实现。该子程序得到输出结果后便会终止。

2. 确定性

算法中的每一步都必须是精确定义的，不能模棱两可。即每一步应该执行哪种动作必须

是清楚的，无歧义的。否则，这样的算法是无法执行的。

例如，算法中不能出现“计算 $x \div 0$ ”这样的描述，因为计算的结果是不确定的。也不能出现“把整数 10 赋给 x 变量或 y 变量”这样的描述，而必须指出，10 赋给某一个变量，或者什么情况下赋给 x 变量什么情况下赋给 y 变量。

3. 数据输入

输入是算法执行过程中需要用到的原始数据，它们取自特定的对象集合。一个算法可以有一个或多个输入，也可以没有输入。

数据输入是编制通用算法的一种手段。数据输入操作可以用赋值操作来代替，但这将降低算法的通用性。例如，更相减损法是求两数的最大公约数的通用算法，在算法的执行过程中，用户输入的两个数据（自然数 m 和 n ）只要是自然数，都可以得到预期的结果。如果将这种输入操作用赋值操作来代替，就变成了一个只能计算两个指定自然数的最大公约数的专用算法了。

4. 信息输出

算法是用来解决给定问题的，所以一个算法必须在执行之后，输出设计者所关心的信息。这就是说，一个算法至少有一个已获得的有效信息输出。例如，更相减损法在执行之后，会将按输入的两个自然数计算得到的最大公约数显示或打印出来。

计算机要解决的实际问题是多种多样的，因而，算法的输出也可以是数字、文字、图形、图像、声音、视频信息，以及具有控制作用的电信号等多种信息形式。

5. 可行性

算法中的任何一步操作都必须是可以执行的基本操作，换句话说，每一种运算至少在原理上可由人用纸和笔在有限的时间内完成。例如，更相减损法就具备可行性。因为它所涉及的两个整数相除、一个整数是否为零、置一个变量的值为另一个变量的值等基本操作，显然都是人能完成的。又如，不限制长度的实数的算术运算则不是可行的，因为某些实数值只能由无限长的十进制展开式来表示，这样的两个数相加不符合可行性的要求。

综上所述，可以得出如下的算法定义：

算法是一个过程，这个过程由一套明确的规则组成，这些规则指定了一个操作的顺序，以便在有限的步骤内提供特定问题的解答。

应该指出，算法的设计工作是一种不可能完全自动化的工作，学习算法的目的主要是学习已被实践证明行之有效的一些基本设计策略。这些策略不仅能用于程序设计，而且对于整个计算机科学技术领域，乃至运筹学、电气工程等其他领域，都是非常有用的。可以预见，一个人如果掌握了这些策略，他的程序设计能力以及整体的分析问题、解决问题的能力将会大大地增强。

1.1.3 算法的基本要素

算法的基本要素有两种：第一，数据对象的运算和操作；第二，算法的控制结构。

1. 算法中数据的运算和操作

通常，计算机中可以执行的基本操作是由计算机能够识别和执行的指令来描述的。一个计算机系统能够执行的所有指令的集合称为计算机的指令系统。所谓程序设计，就是按照解题要求，从计算机指令系统中选择一系列合适的指令构成解题的程序（操作序列）。在一般计

算机系统中，基本的运算和操作有以下4类。

- (1) 算术运算：包括加、减、乘、除等运算。
- (2) 关系运算：包括大于、小于、等于、不等于等运算。
- (3) 逻辑运算：包括与、或、非等运算。
- (4) 数据传输：包括赋值、输入、输出等操作。

算法的设计不受计算机运行环境的影响，一般是按照解题要求，从这四种基本操作中选择合适的操作组成解题的操作序列。算法的主要特征在于算法的动态执行，算法使用一系列最基本的操作，通过对已知条件的逐步加工和变换来实现解题目标，这有别于传统的静态描述或按演绎方式求解问题的过程。传统的演绎数学以公理系统为基础，通过有限次推演来求解问题，每次推演都对问题进行进一步的描述，如此不断推演，直到能够将问题直接描述出来为止。

2. 算法的表示

为了描述一个算法，可以采用多种便于描述的习惯或方式。常用的描述方法有自然语言、伪代码、流程图等。如果一个算法是采用计算机能够理解和执行的语言来描述的，它就是程序。设计这样的算法的过程就叫做程序设计。

(1) 自然语言表示

自然语言就是人们日常生活中使用的语言，可以是汉语、英语或其他语言。例如，中国古算书中的算法（如更相减损法）一般都是用汉语来描述的。这种描述方法通俗易懂。但自然语言的含义往往不太严格，容易出现歧义，也不便描述包含分支部分和循环部分的算法。

(2) 伪代码表示

伪代码是介于自然语言和计算机程序设计语言之间的特殊语言。前面所见到的算法就可以认为是用一种特定格式的伪代码表示出来的。

伪代码是专门表示算法的语言，比自然语言使用方便，可以将算法表示得更加清楚。而且，可以采用一种类似于程序设计语言的伪代码来描述算法，这样的算法也很容易转化为程序。但因为难于找到一种能够被大家普遍接受的伪代码，因而限制了它的使用。

(3) 流程图表示

流程图是用于描述算法的特殊图形，它由一些描述不同种类的操作的带有说明性文字的图框和描述各操作之间执行顺序的流线构成。流程图可以形象地描述算法中各步操作的具体内容、相互联系和执行顺序，直观地表明算法的逻辑结构，是使用最多的算法表示法。

3. 算法的流程图表示

流程图形象地描述了算法中各步操作的具体内容、相互联系和执行顺序，直观地表明了算法的逻辑结构。按它对算法描述的详尽程度，可以分为粗流程图（或称抽象级流程图）和细流程图（或称实现级流程图）。前者可以集中地概括算法的总体逻辑结构，后者可以详尽到与某种程序设计语言相互对应，经过简单地翻译即可形成顺畅的程序。流程图不仅可以用来指导程序设计，而且可以用来辅助程序调试，同时还是人们学习、掌握、交流、移植、改进他人程序的工具。

【例 1-3】 一列火车在某地时的速度为 $V_0 = 40 \text{ km/h}$ ，以加速度 $a = 1.5 \text{ m/s}^2$ 加速行驶，计算 2 分钟后的速度 V 和距开始点的距离 S 。另起一行从物理学可知，本题要用到的公式如下：

$$V = V_0 + at$$

$$S = V_0 t + \frac{1}{2} a t^2$$

流程图有传统流程图和盒图 (N-S 结构流程图) 之分。构成传统流程图的主要功能框如图 1-1 所示。

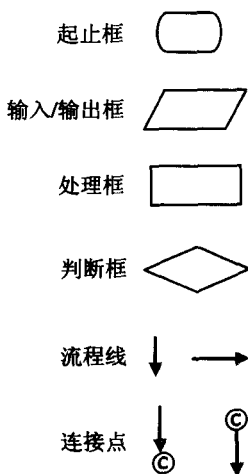


图 1-1 流程图的功能框

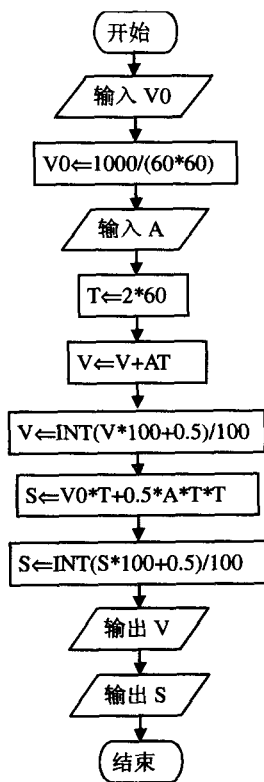


图 1-2 计算速度和距离的算法

在算法中，输入时间、初速度和加速度，即可按公式求出速度和距离，但要解决以下两个问题：

(1) 需要将题目中给定的单位换算成国际标准单位，换算公式为：

$$T(s) = t(h) \times 60$$

$$V(m/s) = V(km/h) \times \frac{1000}{60 \times 60}$$

(2) 如果要求运算结果精确到小数点后第二位，如何进行四舍五入？假设有一个 Int 函数，它的功能是将实数（带小数的数字）的小数部分四舍五入后变成整数。那么可以使用以下表达式对变量 V 的值取二位小数并对第三位四舍五入：

$$\text{Int}(v * 100.0 + 0.5) / 100.0$$

本例的算法流程图如图 1-2 所示。

【注】以下将按程序设计的习惯，将表达式中的实数写成实数形式，并在流程图中以：

变量 ← 表达式

的格式表示赋值操作。