高校非计算机专业计算机教材丛书

FORTRAN 语言程序设计

王肇荣　姚全珠　董　宏　编

COMPUTER

西安电子科技大学出版社

高校非计算机专业计算机教材丛书

FORTRAN 语言程序设计

王肇荣　姚全珠　董　宏　编

西安电子科技大学出版社
1995

(陕)新登字 010 号

内 容 摘 要

本书以国家标准 GB 3057－82"程序设计语言 FORTRAN"为依据，详细地介绍了 FORTRAN77 的基础知识和程序设计方法。全书有大量的例题，其中既有数值计算的，也有非数值计算的。每章都安排了习题，并在书后给出了参考答案。本书所有的例题与习题都在 Microsoft FORTRAN 5.00 编译系统上进行了验算与通过。

本书可作为普通高校有关专业的教材或供计算机应用人员参考使用。也可作为成人教育和职业培训的教学参考用书。本书既便于教学也适合自学。

高校非计算机专业计算机教材丛书
FORTRAN 语言程序设计
王肇荣　姚全珠、董　宏　编
责任编辑　马紫惠

西安电子科技大学出版社出版发行
西安空军工程学院印刷厂印刷
新华书店经销
开本 787×1092 1/16　印张 12 12/16　字数 298 千字
1994 年 11 月第 1 版　1995 年 4 月第 2 次印刷　印数 3 001—11 000

ISBN 7－5606－0350－5/TP・0134　　定价：11.00 元

陕西高校非计算机专业学生
计算机应用知识与应用能力等级考试
专家委员会名单

顾问：	胡正家	西安交通大学教授
委员：	冯博琴	西安交通大学教授
	张遵濂	西北工业大学教授
	罗昌隆	西安电子科技大学教授
	卞　雷	西北大学教授
	曹豫莪	陕西师范大学副教授
	魏文郁	西安冶金建筑学院副教授
	王肇荣	西安理工大学教授
	孙明勤	西北农业大学副教授
	陈　康	西安医学院高级工程师
	李能贵	西安交通大学教务长　教授
	鲍国华	西北工业大学副教授
	肖兴民	西北大学副研究员
	李汝峰	西安电子科技大学副研究员
	孙　朝	省教委高教处副处长

序　言

在普通高等院校中对学生的计算机基础知识与应用能力的培养已经成为各学科各专业教学计划的重要组成部分。高等院校本、专科毕业生的计算机基础知识与应用能力的水平也已成为绝大多数用人单位选择录用人员的重要依据之一。我省各高等院校多年来在计算机基础课程的教学方面进行了精心的组织，工作在计算机基础课程教学第一线的广大教师、工程技术人员和实验室工作人员呕心沥血做了大量艰辛的工作。不过，由于人力、教材、计算机设备等等各方面条件的制约，各院校之间在计算机基础课程教学方面的进展还是不很平衡的。

为了把我省普通高等院校的计算机基础课程教学提高到一个更高的水平，陕西省教委非常重视普通高校计算机基础课程教学的课程建设和各学科各专业学生的计算机知识结构的研究和组织工作。在当前，教材建设又是一项当务之急的基础建设工作。为了使各院校在有关课程的教学内容方面能有一个比较合理、一致的基本要求，省教委邀请了省内近十所高校中工作在有关课程教学第一线的具有丰富教学经验的专家、教授经过认真切磋并参照国家教委工科计算机基础课程教学指导委员会制定的有关课程的教学基本要求，编写了一套“陕西省普通高校非计算机专业计算机教材丛书”。这套丛书包括：《计算机应用基础》(文科类)；《计算机应用基础》(理工科类)；《BASIC 语言程序设计》；《FORTRAN 语言程序设计》；《PASCAL 语言程序设计》；《COBOL 语言程序设计》；《C 语言程序设计》；《微机原理及接口技术》；《计算机软件基础》共九种教材。这不仅是一套适合普通高等院校作为有关课程教学使用的教材，也可作成人教育及各种专门培训班组织的有关课程教学之用，当然也可作为社会上各行各业有关人员学习计算机基础课程的自学教材。这套教材普遍的特点是内容规范、取材精炼、便于组织教学和学生自学。我们相信，这套教材在大面积的使用过程中经过不断的听取意见和锤炼修改，定会成为一套受广大读者欢迎的好材料。

胡正家

1994.2.10

前　言

本书以我国国家标准 GB 3057－82“程序设计语言 FORTRAN”(1982－05－12 发布，1983－05－01 实施)为依据，按照陕西省高校非计算机专业计算机应用知识和应用能力等级考试大纲的要求，介绍 FORTRAN77 语言及其程序设计方法。在学习本书之前要求先掌握《计算机应用基础》(理工科类)一书的内容。

本书编写力求做到概念准确、由浅入深、循序渐进、前后呼应，把 FORTRAN77 的基本内容介绍给读者。全书有大量的例题，其中既有数值计算的也有非数值计算的。每章都安排了习题，并在书后给出了参考答案。本书所有的例题与习题都在 Microsoft FORTRAN 5.00 编译系统上进行了验算与通过。

全书共分十章，第一、二章由王肇荣编写，第三、四、五章由董宏编写，其余五章由姚全珠编写。全书由王肇荣统编和定稿。

由于我们水平有限，书中的缺点和不足，甚至可能存在错误，敬请读者批评指正。

本书由陕西师范大学曹豫莪教授审稿。我们认真研究了曹教授在审稿时提出的宝贵意见，并对原稿做了多处修改，在此向曹豫莪教授表示深切的谢意。

编者

1994 年 3 月

于西安

目　录

第 1 章

FORTRAN 语言概述

1.1 发展概况

FORTRAN 是“FORmula TRANslation”(公式翻译)的字首组合词。它是 1954 年被提出来的，1956 年美国首先在 IBM 704 型计算机上实现了 FORTRAN Ⅰ语言的编译程序。

FORTRAN 是目前国际上广泛流行的一种高级程序语言，最初是为科学计算而设计的，至今仍然主要用于求解数学、工程与科学方面的问题。但是，该语言实际上并无什么地方要强施这一专门性，目前也用于非数值计算，例如用于管理上和商业上。随着 FORTRAN 语言的发展，它的用途也更为广阔。当前，FORTRAN 广泛地作为向学生教授计算机应用和程序设计的一种工具，因为它不用太高深的数学基础且很容易掌握。

FORTRAN 语言问世 40 年来发展很快，先后推出了不同的版本。1966 年美国国家标准协会(American National Standards Institute，简称 ANSI)公布了两个 FORTRAN 标准文本：

美国国家标准 FORTRAN　X 3. 9—1966(习惯上称为 FORTRAN Ⅳ)。

美国国家标准基本 FORTRAN　X 3. 10—1966(习惯上称为 FORTRAN Ⅰ)。

1972 年国际标准化组织(International Standardization Organization，简称 ISO)在美国 FORTRAN 标准文本的基础上，稍加修改后公布了 ISO　FORTRAN 标准，即《程序设计语言 FORTRAN　ISO　1539 - 1972》，它分为以下三级：

基本级 FORTRAN(相当于 FORTRAN Ⅱ)。

中间级 FORTRAN(介于 FORTRAN Ⅰ和 FORTRAN Ⅳ之间)。

完全级 FORTRAN(相当于 FORTRAN Ⅳ)。

1976 年 ANSI 对 FORTRAN X 3. 9—1966 进行了修订，在功能上做了许多改善和扩展，公布了一个 FORTRAN 新的标准草案。1978 年 4 月由 ANSI 正式公布作为新的美国国家标准，称为美国国家标准 FORTRAN　X 3. 9—1978(习惯上称为 FORTRAN77)。1980 年，FORTRAN77 被接受为国际标准，即《程序设计语言 FORTRAN　ISO　1539 - 1980》。

1982 年 5 月 12 日，我国公布了中华人民共和国国家标准 GB 3057—82“程序设计语言 FORTRAN”，并于 1983 年 5 月 1 日开始实施。

为了适应社会发展的需要，FORTRAN 语言还在不断发展中，1985 年 8 月推出了 Microsoft FORTRAN77 V3.31，1989 年推出了 Microsoft FORTRAN5.00。随着版本的不断更新，FORTRAN 语言的功能越来越强。例如，Microsoft FORTRAN5.00 版本，提供了丰富的图形库，支持 OS/2 系统；提供了丰富的接口，可以方便地进行混合语言编程，提供了一些新的数据结构和若干新的控制结构，这些使得 FORTRAN 语言日臻完美。现在 FORTRAN90 已经开始应用。

本书将以国家标准 GB 3057—82“程序设计语言 FORTRAN”为依据来介绍 FORTRAN77 语言及其程序设计方法。全书的例题与习题都在 Microsoft FORTRAN 5.00 编译系统上进行了验算与通过。Microsoft FORTRAN 符合 ANSI X3.9—1978 标准的程序设计语言 FORTRAN77。

一般来说，目前计算机上的 FORTRAN 文本是不违背标准文本的。但是，各种机型的硬件(特别是输入和输出)有所不同，以及相应的编译系统实现的方法也不同。因此，在不同计算机上实现的 FORTRAN 语言可能会有一些具体规定或少量的修改。所以，读者在计算机上使用 FORTRAN77 之前，最好先阅读该机上的 FORTRAN 文本，尤其要注意那些与标准文本不一致的地方。

1.2 FORTRAN 源程序基本结构

一、FORTRAN 字符集

FORTRAN 字符集由 26 个字母，10 个数字和 13 个特殊字符组成。

1. 字母

字母是下列 26 个字符之一：

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

2. 数字

数字是下列 10 个字符之一：

0 1 2 3 4 5 6 7 8 9

对数字串要解释为数值时，应解释为十进制数。

3. 特殊字符

特殊字符是下列 13 个字符之一：

	空格(本书中用␣表示)	)	右括号
=	等号	,	逗号
+	加号(正号)	.	小数点
−	减号(负号)	'	撇号
*	星号(乘号)	:	冒号
/	斜线(除号)	$	币号
(	左括号		

在 FORTRAN77 源程序中，除一些特殊规定的情况外，一般只使用这 49 个字符，而且字母的大小写不予区分，空格符也无意义。但对特殊规定的一些情况，空格符和字母大小写均有意义，这在后面遇到具体情况时将予介绍，而且在这些情况中使用的字符可以超出上述字符集的范围，当然以处理系统能接受为限。

二、源程序的书写格式

用 FORTRAN77 语言编写的程序称为 FORTRAN77 程序，又称为源程序。FORTRAN 源程序必须严格地按照一定的格式书写。FORTRAN 源程序一行最多可以有 80 个字符。行中的字符位置称为列，一行中各列从左至右依次连续编号为 1 到 80。这 80 列分为四个区，分别书写不同的内容。第 1 至第 5 列为标号区，用来给语句标上标号；第 6 列为续行标志区；第 7 至 72 列为语句区，第 72 列至 80 列为注释区。

在一行的语句区中最多只能写 1 个 FORTRAN 语句。当一个语句在一行中写不下时，可以在下一行的语句区中继续写。称一个语句的第 1 行为始行，它的其它行称为续行。FORTRAN 规定一个语句不能多于 19 个续行，也就是说一个语句不能包含多于 1320 个字符。始行的第 1 列至第 5 列上可含有语句标号或全是空格符，而第 6 列必须是空格符或数字 0；续行的第 1 列至第 5 列必须全是空格符，而第 6 列上必须是除空格符或数字 0 以外的 FORTRAN 字符集的任意字符。始行与它的续行必须书写在连续的行上，中间不允许插入除注解行之外的任何其它语句行。

若一行上的第 1 列是字母 C 或星号 *，则表示该行为注解行。注解行的内容写在第 2 至第 72 列上。注解行的内容可以由处理系统能接受的任何字符组成。注解行完全不影响可执行程序，注解行可用于对源程序作一些说明，以增强源程序的可读性。注解行可以出现在源程序的任何地方，可置于一个语句的始行之前，也可置于始行和第一个续行之间，或两个续行之间。

语句标号提供引用语句的标志，或作为控制语句转向的目标。因此，FORMAT 语句(见 3.2 节)和作为控制语句转移到达的语句必须有标号，其它语句的标号可有可无，并不影响程序的执行。语句标号书写在始行的标号区中，可以由 1 到 5 位数字组成，可以是 1 到 99999 的无符号整数。在区分语句标号时，空格或前导零没有意义。例如，下列标号

␣␣150　　1␣5␣0　　0150␣　　150　　00150

编译系统认为是相同的标号。在程序中，语句标号出现的次序与它的数值大小无关。在一个程序单位中，同一个语句标号不得标识一个以上的语句。除程序执行控制转向外，程序是按可执行语句在程序单位中出现的次序执行的，与标号值的大小或出现与否均无关。

三、源程序的基本结构

1. 程序单位

一个可执行的 FORTRAN 程序由一个或多个程序单位组成。程序单位由语句和任选注解行的序列组成。一个程序单位或者是一个主程序或者是一个辅程序。一个程序单位的语句标号仅在该程序单位内有效。程序单位必须以 END 语句结束。

2. END 语句

END 语句的形式是

END

END 语句是可执行语句，它一定是程序单位的最后一个语句，它表明一个程序单位的语句和注解行序列的结束。它必须写在始行的第 7 列到第 72 列上，END 语句不允许有续行。

如果主程序的 END 语句被执行，则整个程序执行就结束；如果函数辅程序或子程序辅程序的 END 语句被执行，则控制返回引用程序。在一个程序单位中不允许有 2 个或 2 个以上的 END 语句。

3. 主程序

主程序一般是以 PROGRAM 语句开头，以 END 语句结束的一个程序单位。PROGRAM 语句的形式是

PROGRAM name

其中 name 是出现 PROGRAM 语句时主程序的符号名，它不应与其它被使用的名字相同。

主程序命名并不是必要的，所以主程序中 PROGRAM 语句可有可无。但是，如果要写 PROGRAM 语句的话，就必须是主程序的第一个语句，而且要起一个名字。

4. 辅程序

辅程序包括下列 3 种：

函数辅程序，它以 FUNCTION 语句开头，END 语句结尾。

子程序辅程序，它以 SUBROUTINE 语句开头，END 语句结尾。

数据块辅程序，它以 BLOCK DATA 语句开头，END 语句结尾。

一个可执行的 FORTRAN 程序，有且仅有 1 个主程序，但可以有零个到多个辅程序。程序的执行是从主程序的第一个可执行语句开始的。

1.3 程序流程图

程序流程图又称为框图，它是用一些图框来表示各种类型的操作，以便形象地描述解决问题的步骤。

美国国家标准协会 ANSI 规定了一些常用的流程图符号，见图 1.1。

例 1.1　某房地产公司有两块长方形的地皮。第一块宽度为 95.8 m，长度为 223.5 m。第二块宽度为 198.6 m，长度为 219.6 m。试画出计算每块地皮面积和两块地皮总面积的流程图。

图 1.2 就是满足上述要求的流程图。

例 1.2　画出求解一元二次方程 $Ax^2+Bx+C=0$ 的流程图。见图 1.3。

有些算法画出的流程图会较复杂，为了避免流程线的交叉或过长，常常使用连接点，可以使流程图清晰易读。图 1.4 实现的是一个完整的算法，我们分三块画出。

图形符号	名称	意、义
	起止框	表示流程的开始、终止或暂停。
	输入/输出框	表示输入或输出数据。
	处理框	表示一般的处理操作。
	判断框	表示要做出判断。在几条可供选择的路径中，根据条件选取其中一条路径。
	流程线	表示流程的去向。
	连接点	表示流程图中流程线的连接点。

图　1.1

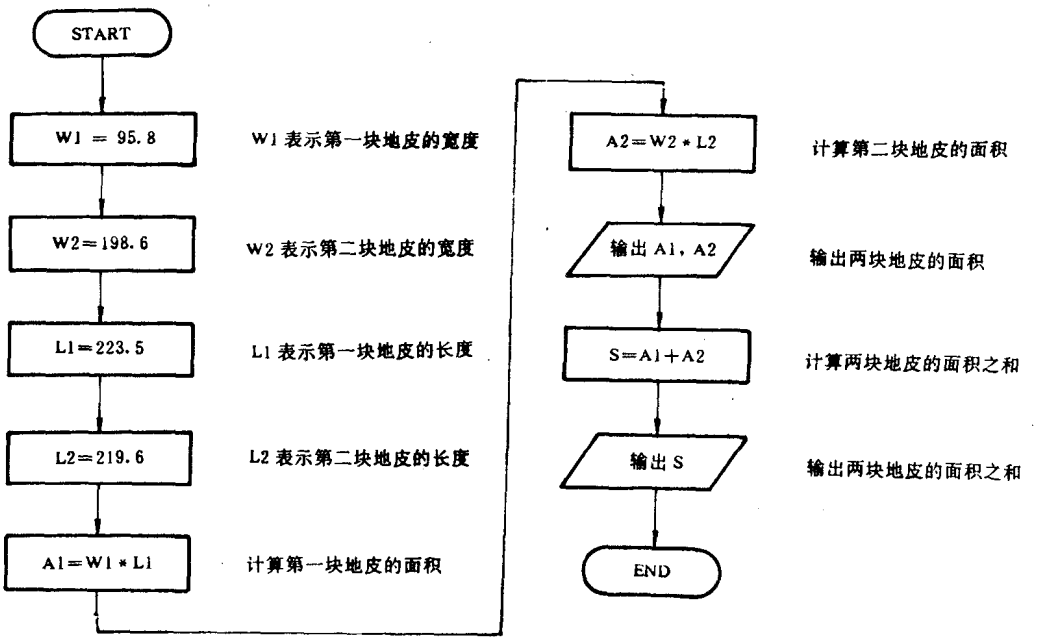

图　1.2

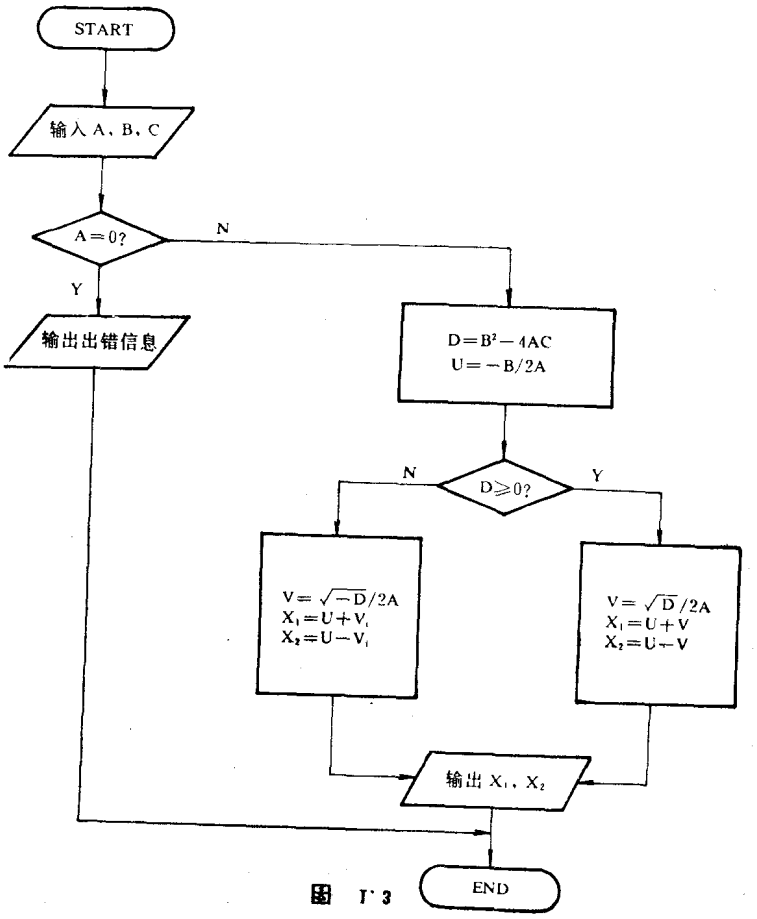
START
输入 A，B，C
A=0?
N
Y
输出出错信息
D=B²−4AC
U=−B/2A
N
D≥0?
Y
V=√−D/2A
X₁=U+V
X₂=U−V
V=√D/2A
X₁=U+V
X₂=U−V
输出 X₁，X₂
END

图 1·3

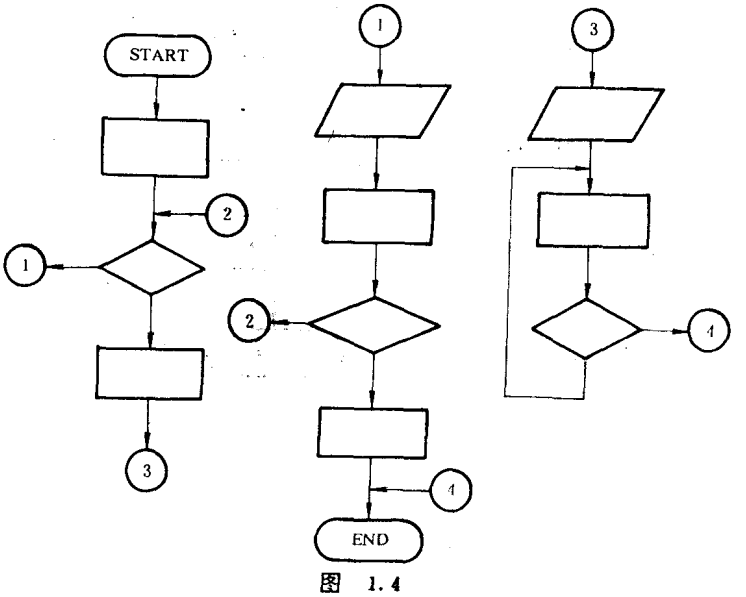
START
1
2
3
1
2
3
1
1
END

图 1.4

1.4　结构化程序设计

一、结构化程序设计思想的产生

20 世纪 50 年代到 60 年代，一个程序员在掌握了程序设计语言和程序设计中的一些基本技巧之后，只要会使用一些算法并经过一定的实践，就可以编出满足当时要求的程序来。当时编出的程序与程序员本人的业务水平关系极大，程序中使用的技巧越多越巧妙，运行速度就越快，所占用的存贮单元就越省。因此，人们常称程序设计既是一门科学也是一门艺术。从 60 年代末期，越来越多的大型科研课题使用计算机，并陆续出现了大型软件系统，如操作系统、数据库等，这些都给程序设计带来了新问题。一个大的程序要花费众多的人力，有时需要几千人年的工作量，因而不能由一个人或少数几个人来编制。但是，当许多人分别编制的风格各异的程序连结在一起的时候，常常是可靠性差、错误多，维护和修改都很困难。隐藏的错误也不易被发现和纠正。为此，就促使人们开始研究程序设计中的一些最基本的问题：什么是程序的基本组成部分？应该用什么样的方法来设计程序？程序设计的主要方法和技术应如何规范化和工程化，等等。

1969 年，荷兰数学家 Dcjkstra 首先提出了结构化程序设计的概念。他强调了从程序结构和风格来研究程序设计。一个程序结构好，是指程序结构清晰、便于编写、便于阅读、便于验证、便于修改和维护。结构好的程序从效率上看，并不一定是最好的程序。这里所指的效率是从时间和空间两个角度来衡量，即希望编出来的程序越短越好，运行速度越快越好。但是，结构好的程序能减少程序出错的机会、提高程序的可靠性和保证程序的质量。随着硬件技术的发展，当前计算机运行速度大大提高，存贮容量也很大了，因而程序的可靠性和可维护性已成为第一要求。除了系统的核心程序及其它一些有特殊目的(如军事目的)程序外，在通常情况下，宁可降低效率，也要保证程序有好的结构。从此，人们开始研究、总结出一套程序设计的基本原理和方法，使程序设计尽可能减少对程序员个人的风格、技巧的依赖，而逐渐上升为一门科学性的学科。当然程序设计仍然是由程序员来完成的，还是保持着一定程序艺术的特点。目前，普遍使用的和比较成熟的程序设计方法，是结构化程序设计方法。

二、结构化程序设计方法

结构化程序设计方法主要包括以下两个方面的内容。

1. 模块化程序设计

程序中完成一种特定功能的指令集合称为模块。模块化程序设计的基本思想是在进行系统设计时，对总体方案采取自上而下、逐步细分的方法，把一个大而复杂的问题分解为若干个相对独立、功能单一的模块。继之，是对这些单一功能的模块的研制，因而使复杂的研制工作简化了。由于模块的相对独立性也能有效地防止错误在模块之间扩散蔓延，当许多模块连结在一起时，减少了相互之间的干扰和影响，从而提高了系统的可靠性。

考虑了总体方案的程序设计的结构之后，就对每个模块编制程序。那么，整个程序就由一些相对小的程序组成，而这些程序都具有一定的相对独立性，可以分别调试和修改。

这样，一方面可以使许多人分头工作，缩短软件研制的周期；另一方面也使整个系统思路清楚，便于发现问题并及时纠正。一般地，整个系统的模块可用树形图表示。“树根”是主控模块，下连第一层子模块，各个子模块下还可以有更下一层子模块，……直到最底层模块。最底层模块称为“树叶”。图 1.5 表示一个三层的模块结构。

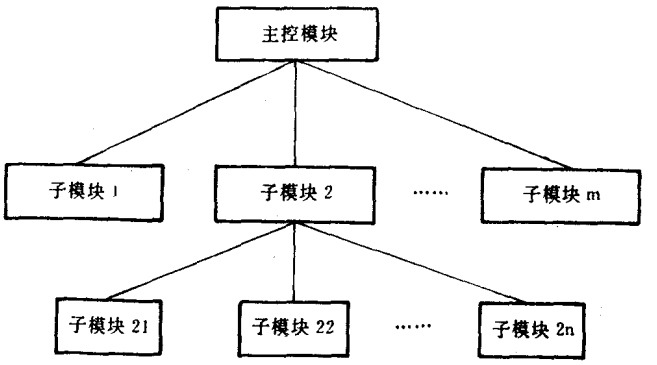

图　1.5

采用模块化程序设计的优点是：

①由于模块相对独立，可以对每一个模块独立编制、调试和修改程序。这种修改只会对本模块发生影响，而不会产生涉及其它模块的连锁修改。

②由于模块本身只有一个入口和一个出口，于是模块内部的数据结构只有模块内部提供的操作才能给以改变，因而保证了模块内部数据结构的安全性和完整性。

③由于各模块只在上下级之间有接口，同层次之间无接口，因而程序结构清晰，模块的正确性就容易保证。

2. 三种基本结构

这里所说的结构化程序设计是指采用三种基本结构，就能使程序具有好的结构。

1966 年，Böhm 和 Jacopini 证明了程序设计语言中只要有三种基本结构，就足以表示出各种各样的其它形式的结构。这三种基本结构是顺序结构、选择结构和循环结构。

(1) 顺序结构

如图 1.6 所示，虚线框内是一个顺序结构。顺序地执行语句序列 A 和 B，只有当 A 执行完成之后才执行 B。顺序结构是最简单的一种基本结构。

(2) 选择结构

如图 1.7 所示，虚线框内是一个选择结构。此结构必包含一个判断框，当条件成立(YES，用 Y 表示)或不成立(NO，用 N 表示)时分别执行语句序列 A 或 B，二者择其一。不管执行哪一个语句序列，执行结束后都转移到同一出口的地方。A 或 B 两个语句序列中可以有一个是空的，即不执行任何操作。如图 1.8。

(3) 循环结构

循环结构又称为重复结构，即反复执行某个语句序列。这里介绍两类循环结构：

1) 当型(loop - while 型)循环结构

见图 1.9，虚线框内是一个当型循环结构。它的功能是：当给定的条件成立时，执行语句序列 A(在循环结构中语句序列 A 称为循环体)，执行完 A 后，再判断条件是否成立，如果仍然成立，再执行循环体，如此反复执行循环体，直到某一次条件不成立时而离开此循

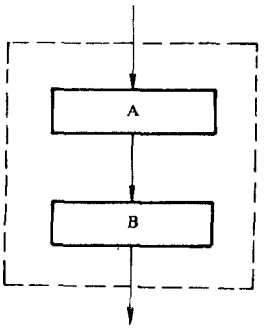

图　1.6

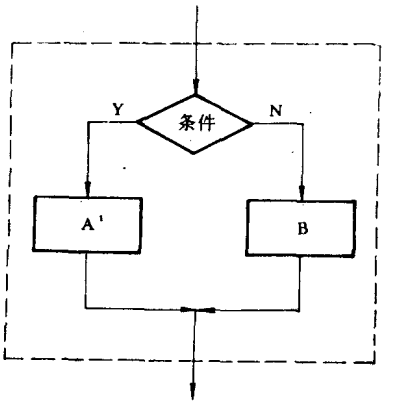

图　1.7

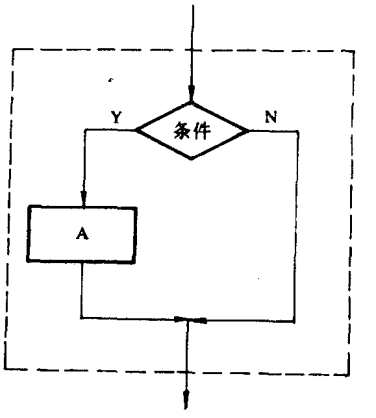

图　1.8

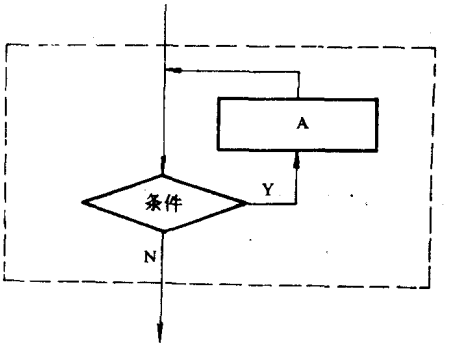

图　1.9

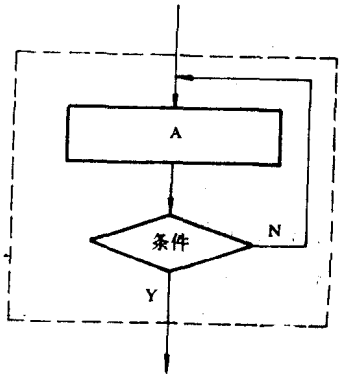

图　1.10

环结构。

2）直到型(loop - until 型)循环结构

见图 1.10，虚线框内是一个直到型循环结构。它的功能是：先执行语句序列 A，即执行循环体，然后判断给定的条件是否成立，如果条件不成立，则再执行 A，然后再对条件进行判断，如果条件仍然不成立，又执行 A，……如此反复执行 A，直到给定的条件成立时而离开此循环结构。

图 1.11 和图 1.12 分别是当型循环和直到型循环的应用例子。

图 1.11 和图 1.12 的作用都是打印 10 个数 1，2，3，…，10。可以看出对同一个问题既可以用当型循环来处理，也可以用直到型循环来处理。

现在我们来分析一下这两种循环结构的异同。

①两种循环结构都能处理需要重复执行的操作。

②当型循环是一种先判断后执行的循环结构，当条件不成立时停止循环，循环体可能一次也不被执行；直到型循环是一种先执行后判断的循环结构，循环体至少被执行一次，当条件成立时停止循环。

③对同一个问题，如果分别用当型循环结构和直到型循环结构来处理，则两者结构中