

FORTRAN

结构程序设计基础

王朝宝 编

北京航空學院出版社

FORTRAN

结构程序设计基础

王朝宝 编

北京航空學院出版社

内 容 简 介

结构化程序设计思想和方法是当今世界上公认的保证程序设计的正确性、可靠性的最先进的思想与方法。FORTRAN又是国内外广泛用于科技计算的一种计算机高级语言。把二者有机的结合起来,使初学计算机高级语言的读者,从开始就为养成良好的程序设计风格与习惯打下基础。这就是本书的宗旨。本书编写的形式较新颖,根据所学内容的难易,重点与非重点等分为:自学部分;讲解部分;深入部分。语言通俗易懂,适于自学。

本书可供理工科非计算机专业本科生、大专生教科书或教学参考书。

FORTRAN结构程序设计基础

编 者 王朝宝

责任编辑 马晓虹

北京航空学院出版社出版

新华书店北京发行所发行 各地新华书店经售

昌平振兴胶印厂印装

印张:15.25 字数:377.5千字

1987年11月第一版 1987年11月第一次印刷 印数:5000册

ISBN7-81012-033-6/TP-010 定价:2.6元

目 录

前 言

第一章 绪论

§ 1—1	本课程的重要性及特点	(2)
§ 1—2	电子计算机概述	(3)
§ 1—3	计算机语言	(6)
§ 1—4	FORTRAN语言发展概况	(9)
§ 1—5	用高级程序设计语言解题的全过程	(9)
§ 1—6	FORTRAN程序举例	(11)
§ 1—7	结构化程序设计	(17)
§ 1—8	有关计算机术语的介绍	(18)
习 题		(22)

第二章 FORTRAN基本术语

§ 2—1	FORTRAN语言的基本字符集和语言元素	(23)
§ 2—2	FORTRAN程序的书写	(26)
§ 2—3	FORTRAN程序的结构	(29)
习 题		(31)

第三章 常数、变量及其变量类型说明

§ 3—1	常数	(32)
§ 3—2	变量及类型说明	(36)
§ 3—3	标准函数	(39)
习 题		(40)

第四章 输入/输出初步

§ 4—1	概述	(42)
§ 4—2	读/写语句	(43)
§ 4—3	格式语句	(44)
§ 4—4	字段描述符	(45)
§ 4—5	字段描述符组	(53)
第四章 [附] (含第一次上机习题)		(53)
习 题		(56)

第五章 赋值语句与控制语句

§ 5—1	算术赋值语句	(58)
§ 5—2	逻辑赋值语句	(64)

§ 5—3 控制语句·····	(66)
§ 5—4 无条件转语句·····	(67)
§ 5—5 条件语句·····	(69)
§ 5—6 计算转语句和赋值转语句·····	(79)
§ 5—7 停语句与暂停语句·····	(83)

第五章[附](含第二次上机习题)·····	(84)
习 题·····	(84)

第六章 数组与循环

§ 6—1 数组与维数语句·····	(88)
§ 6—2 利用条件语句实现循环的类型·····	(94)
§ 6—3 循环语句与继续语句·····	(97)
§ 6—4 多重循环·····	(104)
§ 6—5 循环的控制转移·····	(106)
§ 6—6 程序举例·····	(109)
第六章[附](含第三次上机习题)·····	(119)
习 题·····	(120)

第七章 输入/输出

§ 7—1 无格式读/写语句·····	(122)
§ 7—2 输入/输出表·····	(123)
§ 7—3 格式说明和输入/输出表的相互作用·····	(126)
§ 7—4 字段描述符·····	(132)
§ 7—5 打印机纵向走纸控制·····	(137)
§ 7—6 打印曲线及数组中的格式说明·····	(138)
§ 7—7 辅助的输入/输出语句·····	(144)
第七章[附](含第四次上机习题)·····	(145)
习 题·····	(145)

第八章 函数和辅程序

§ 8—1 概 述·····	(148)
§ 8—2 语句函数·····	(150)
§ 8—3 外部函数及函数辅程序·····	(153)
§ 8—4 子程序辅程序·····	(166)
§ 8—5 辅程序举例·····	(170)
§ 8—6 过程总结·····	(174)
§ 8—7 可调数组与通用程序·····	(176)
第八章[附](含第五次上机习题)·····	(180)

~ 2 ~

习 题.....	(181)
第九章 数据联系语句与数据初值语句	
§ 9—1 公用语句.....	(184)
§ 9—2 等价语句.....	(189)
§ 9—3 数据初值语句.....	(195)
§ 9—4 数据块辅程序.....	(197)
习 题.....	(198)
第十章 结构化程序的编制与调试	
§ 10—1 算法及算法的重要性.....	(200)
§ 10—2 程序编制的一般步骤.....	(202)
§ 10—3 结构化程序设计的基本方法.....	(205)
§ 10—4 程序优化的方法.....	(210)
习 题.....	(214)
第十一章 FORTRAN——77与标准FORTRAN的差异	
§ 11—1 概 述.....	(216)
§ 11—2 FORTRAN——77标准中新增加的部分.....	(216)
§ 11—3 对原有语句增加新功能的部分.....	(227)
附 录 标准函数表.....	(233)
参考文献资料.....	(238)

前 言

随着计算机的应用日益广泛,人们对掌握“人类第二文化”——计算机高级程序设计语言重要性的认识也在不断提高。*FORTRAN*语言是自五十年代中期至今国内外科技计算中使用较为广泛的一种高级程序设计语言,也是我院非计算机专业本科生必修的计算机高级程序设计语言。

虽然美国标准化协会ASA于1978年通过的*FORTRAN*—77比1972年国际标准化组织ISO通过的*FORTRAN*具有更强的功能,更多的优越性,并且近年来国内计算机系统软件中具有*FORTRAN*—77编译系统的也逐渐增多。但是本教材在编写时考虑:ISO通过的*FORTRAN*〔完全级相当于美国ASA组织1966年通过的标准*FORTRAN* (相当于*FORTRAN*—IV),本教材中用*FORTPAN*代表〕仍是*FORTRAN*—77的基础,它的限制比*FORTRAN*—77要多;也考虑到目前国内计算机不具备*FORTRAN*—77编译的仍居多数;而且还考虑到*FORTRAN*—77编译系统对*FORTRAN*兼容,即具有*FORTRAN*—77编译系统的计算机用*FORTKAN*所编程序也可运行。因此本教材仍以*FORTRAN*为基本内容。对于*FORTRAN*—77是以与*FORTRAN*的区别形式加以介绍的。这样安排可以使读者学后,能用*FORTRAN*较熟练的编程,若需用*FORTRAN*—77编程则只需读者有一熟练过程即可达到。反之若仅讲*FORTRAN*—77,则读者在只具有*FORTRAN*编译的计算机面前将会束手无策,无所适从。

自从六十年代末期,由著名计算机专家戴克斯特拉创立和随后不断完善的结构化程序设计思想和方法引起了软件设计的革命,使人们对结构化程序设计思想和方法极为重视。尽管*FORTRAN*语言规范不完全符合结构化程序设计的要求,然而现在程序编制人员仍以结构化程序设计思想和方法,用*FORTRAN*语言编出符合结构化程序设计要求的程序,以保证程序的正确性,易读性,提高生产率。为了使读者能基本掌握这种先进的科学的程序设计思想和方法,使读者在学习语言的开始,就打下良好的正确的程序设计风格和习惯,本教材将以结构化程序设计作为主导思想贯彻始终。也正因为如此,本教材名称为*FORTRAN*结构程序设计基础。

学习高级程序设计语言的目的是在掌握一种高级程序设计语言后,能编写上机程序及能正确阅读已有程序,因而读者在使用本教材时切记不要死记硬背语言规范,而忽视了“编”与“读”程序的基本目的。

本教材在编写形式上将不同于一般教材,每章节将标出“自学部分”、“讲解部分”、“深入部分”。以贯彻“讲一、练二、考三”的教改指导方针,积极培养读者的自学能力。

教材每章之后均有习题,每次课后可选做若干题。学习本课程期间要求通过5个上机题,上机题目也在教材中适当部分出现。

本教材适用于高等院校非计算机专业理工科本科生一年级第一学期使用,教学安排为33学时。对于其他年级也可适用。

由于水平及时间仓促,其中不妥和错误之处恐不在少数,恳请读者批评、指正。

第一章 绪 论

本章将向读者强调学习本课程的重要意义及本课程的特点,以便帮助读者更好地学习本课程。同时本章还将电子计算机的有关基本知识;计算机语言特别是FORTRAN语言的发展等作些简单介绍。为使读者对FORTRAN程序有一感性认识,本章专门有一节介绍FORTRAN程序举例。本章最后介绍了当前软件设计的基本方法——结构化程序设计思想和方法出现的背景以及其基本内容,以使读者在今后各章学习时能牢牢抓住这一指导思想。

§1-1 本课程的重要性及特点 自学部分

随着电子计算机在各个领域的应用日益普遍与深入,人们对计算机语言,尤其是高级程序设计语言重要性的认识也逐渐加深与提高。特别是1981年上半年,在瑞士召开的“世界计算机应用年会”上,提出了一个极为重要的新的概念:把高级程序设计语言称为“人类的第二文化”。由此可以看到,高级程序设计语言对现在人类生活的重要意义。1983年9月上旬,“全国高等院校非计算机专业计算机课程讨论会”的会议纪要指出:八十年代我国大学毕业生必须至少掌握一门高级程序设计语言,使他们成为推广计算机应用的生力军。1983年美国最畅销书之一“大趋势——改变我们生活的十个新趋向”的译者前言中写道:“在信息社会中,美国人必须掌握两种语言,英语和计算机语言。……不知道怎样去应用计算机的人处于不利的地位”。

正因为该课程如此重要,以至于全国理工科高等院校均把它作为本科生必修科目。它也是保证本科生大学四年期间,计算机应用不断线的基础。

本课程的特点是:课程内容比较琐碎、死记的东西较多,利用逻辑推理的亦较多。但是在学习本课程中切忌死记硬背,应该紧紧抓住程序设计这个中心,把各种语句功能、规定、语法灵活地通过程序的编制牢固掌握,与此同时还要学会并掌握算法逻辑判定的方法,这对于检查所编程序有无逻辑错误是十分重要的。

其次本课程另一特点是必须多次通过上机算题,才能熟悉和掌握本课程的基本内容。也就是说为了熟悉和掌握本课程的基本内容必须通过实践环节。如果只在纸上编程序而不上机实践,可以肯定本课程的基本内容不会掌握好。为此,在教学过程中必须十分重视和珍惜每次上机的实践。

由于FORTRAN语言的书写有严格的规定,因此在学习本课程中必须注意这一特点,我们的作业和上机前的编程,均要使用特制的FORTRAN程序设计练习本和程序纸。书写格式必须严格按照规定,而且字迹要清楚、整齐,不允许出现涂抹。为了说明FORTRAN程序书写的重要性,举一实例。

美国一次向金星发射飞船时,整个程序中有这样一条循环语句:

```
DO 3 I = 1, 3
```

但因程序员书写时疏忽,写成了

~ 2 ~

DO 3 I = 1, 3

即将1与3之间的逗号，误写成小数点了。看来是无足轻重的笔误，但却将循环语句变成了赋值语句。这样的小错误却导致价值10亿美元的金星计划全盘失败。有人会说，书写错了，找出来改正就行了。要知道对于大型软件来讲，寻找类似这样的错误，就犹如在厚厚一本百科全书里找出一个拼错了的单词一样是极其困难的。因此学习本课程过程中的作业与上机编程要格外地仔细、细心。

§1—2 电子计算机概述

一、电子计算机的应用

自学部分

随着计算机的发展，它的应用日趋广泛与深入。从工业、农业、科研、教育、国防……直到人们的日常生活，处处都显露出它的“才华”。

例如，对加拿大圣劳伦司海路的潮汐(*the tides of the Saint Lawrence seaway in Canada*)的数学计算需要让加拿大全部数学家工作一百年才能完成。然而20多年前由多伦多大学的计算机仅用一周多时间就完成了。如果采用今天的计算机，则只需若干分钟。由此可以看到电子计算机的“才华”。

电子计算机目前可分三大类：

1. 数字计算机；
2. 模拟计算机；
3. 混合式计算机。

概括起来电子计算机主要有以下几方面的应用：

1. 数值计算

这是电子数字计算机应用的一个基本方面。由于它运算速度快；计算准确；自动化程度高；通用性强，因此广泛地用在科研和工程计算上。如导弹弹道计算等。

2. 数据、信息处理和情报检索

利用电子计算机查找情报、资料既快又准。据统计每年国外公开发表的文献达500万种，杂志几万种，找资料是个大问题。若用电子计算机可以很快地帮你找到所需要的资料、情报。银行、企业的财会工作，企业的决策、管理，人口统计，天气预报等亦属于数据、信息的处理。

3. 实时控制

这是电子计算机应用颇为广泛的一个领域。如无人驾驶飞机、汽车；数控机床；自动轧钢系统等均属实时控制。近年来微处理机的发展，更促进了计算机在实时控制方面的广泛应用。

4. 计算机辅助设计

电子计算机可使设计过程趋于半自动化或自动化，既节省人力、时间又降低成本，保证设计质量。如大规模集成电路的辅助设计，利用电子计算机几乎成为必需手段。

5. 人工智能

亦就是利用电子计算机模拟人的智能。它不仅包括只能从事某些繁重、危险的劳动——

机器人，而且还包括现代具有“判断”、“思维”能力的智能机器人，如下棋、某些数学定理的证明等。例如1966年美国利用机器人打捞落入地中海靠近西班牙海岸附近的氢弹就是机器人应用的一例。又如：火星和地球相距最远时达36000万公里，无线电波来回一趟要40分钟。当火星上发生预想不到的事情时，地球20分钟后才可知道，人类做出反应及信号发回又要大于20分钟，显然这是不行的。为此美国研制火星智能机器人，预定1986年发射，在火星上工作两年，作2000公里长途考察，随时代替人类处理突发事件。

虽然电子计算机已进入人类社会生活的各个领域但它并不是万能的、绝对正确的。例如1980年6月2日午夜，北美防空司令部里一台大型计算机的终端显示屏上，突然出现令人毛骨悚然的信息：苏联的洲际导弹及带核弹头的由核潜艇发射的导弹已飞临美国。为此在数秒钟后，美国的110多架B—52战略轰炸机迅速爬出机窝；1054枚民兵和大力神导弹进入一等战斗状态；24艘装有氢弹头的导弹核动力潜艇疾速地沉入到限定的发射弹导的深度……一场核大战大有一触即发之势。3分20秒后，通过全球卫星监视网验证，空中并没有苏联导弹。

这场虚惊的罪魁祸首是计算机中一块硬币大小的元件出了故障。类似的错误还有，所以我们对计算机的应用要有正确的认识与评价。

二、计算机的发展概况

世界上第一台电子数学计算机ENIAC(Electronic Numerical Integrator And Calculator)是由美国的宾夕法尼亚大学为美国陆军炮场用于计算炮击表而研制的。从1943年开始研制到1946年成功，共用了三年多时间。

这台计算机共用了：18000个电子管；1500个继电器；占地160 m²；重达30吨；功耗150KW；耗资几千万美元。然而运算速度仅为5000次/秒。

但是随后的卅多年，电子计算机的发展已达到迅猛的程度。平均5~8年，计算速度就要增加10倍；体积减少10倍；成本降低10倍。如1977年美国生产的intel8048电子计算机，它的芯片大小仅有几mm²；重量仅有几十克；功耗只有3W；价格几十美元。但是运算速度却达10万次/秒。

三十多年来电子计算机的发展经历了电子管；晶体管；集成电路及大规模或超大规模集成电路四代，与此同时软件也有了相应的发展。

这四代的划分及特征如下表：

计算机代	起止年代	代表机器	运算速度 (次/秒)	硬 件		软 件	应 用
				逻辑元件	主存贮器		
第一代	1946 ~1957	IBM-704 UNIV AC-1	几千~ 几万	真空管	磁鼓延迟 线、磁芯	机器语言 汇编语言	科 学 计 算
第二代	1958 ~1964	IBM-7090 ATLAS	几万 ~几十万	晶体管	磁芯	程序设计 语言 管理程序	科学计算 数据处理 事务管理
第三代	1965 ~1970	IBM-360 CDC-6000 PDP-8	几十万 ~几百万	中、小规模 集成电路	磁芯	操作系统 会话式语言	实现系列化 标准化。广泛 应用各领域
第四代	1970 ~目前	IBM-4300 FACOM	几千万	大规模、 超大规模集 成电路	半导体	可扩充语言 数据库	计算机网 络应用 微处理机应用

近几年国外如日本、美国利用超大规模集成电路生产了计算速度高达亿次/秒以上的计算机,最高达6亿次/秒,并开始研制光计算机。

今后计算机的发展趋势,概括起来将是向着:

- (1) 微小型化或超大型化发展;
- (2) 计算机网络方向发展;
- (3) 人工智能;
- (4) 光计算机。

三、电子数字计算机的组成及工作原理

讲解部分

(一) 组成:

现代计算机系统由硬件及软件组成。

1. 硬件

硬件就是机器设备的总称。它主要由五部分组成,即1)运算器;2)存贮器;3)控制器;4)输入设备;5)输出设备。

它们的作用简单介绍如下:

1) 运算器

可以完成各种算术运算、逻辑运算、字符运算。此外它还可以传递信息(如指令、数据等)。通常运算器是由若干个加法器和移位器组成,其指标是运算速度。如IBM—4341计算机运算速度为86万次/秒。

2) 存贮器

其作用是存放数据和程序指令的装置。通常计算机把数据、指令用二进制表示,而存贮器的作用就是按需要存、取它们。存贮器是由大量的存贮单元构成,每一个存贮单元可存贮一个数或一条指令。存贮单元的特点是“取之不竭,一冲就跑”。即是若从某单元中取信息,则取走后,该单元中仍保留其原有信息。若往该单元送入新信息,则将该单元原有信息冲跑即不再存原有信息,新信息占据了原有信息的单元。存贮单元的总数即是机器的存贮容量。该值越大,机器的功能愈强。它是电子计算机主要性能指标之一。例如:

VT—60机存贮量为 $128K=128 \times 1024=131072$ 单元。

FELIX—C256机存贮量为 $256K=262144$ 单元。

IBM—4341机存贮量为 $2M=2 \times 1000000=2$ 百万单元。

[$1K=2^{10}=1024$ 单元; $1M=1$ 兆= 10^6 单元]

由于机器的不同,每个存贮单元由多少个二进制位组成也不相同,有的是八位、有的是十六位、有的是三十二位。通常机器越小存贮单元的二进制位数越少,因此存贮量也有用二进制位表示的。所谓二进制位即是由一个记忆元件如磁芯等构成。

存贮器除存贮容量指标外还有存贮周期的指标。如VT—60机,存贮周期为 $800 \sim 900 \mu s$ 。显然存贮量越大,存贮周期越短,计算机水平越高。

3) 控制器

控制器是电子计算机的指挥系统,它按输入到计算机里的程序指令,向机器各部分发出信号,以使整个计算机按要求协调地工作。

4) 5) 输入设备/输出设备

向(自)电子计算机输入(出)数据、指令,以及文字信息的设备,即称为输入(出)设备。输入设备如读卡机、纸带输入机、终端等。输出设备如宽行打印机、窄行打印机、自动绘图机、终端等。

通常人们把运算器、存贮器、控制器称为主机。其中又将运算器和控制器合称为中央处理机(CPU)。把输入/输出设备、外存贮器、终端设备等称之为外部设备。

2. 软件

硬件加软件方可使电子计算机工作。软件它包括有:

1) 系统软件:如编译程序、操作系统、诊断检查程序等。

计算机没有系统软件是无法工作的。

2) 应用软件:为便于用户使用计算机,通常以软件包形式出售应用软件,如数学计算软件包、绘图软件包、企业管理软件包等。

(二) 工作原理

这五个主要组成部分之间的工作联系如图1—1。

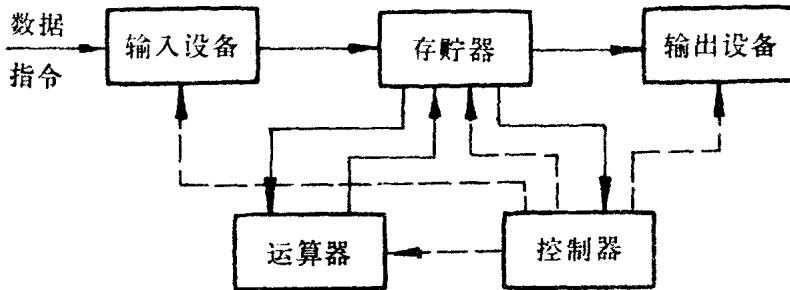


图 1—1

图 1—1 中——→代表信息传输

- - ->代表控制命令

关于电子计算机详细的工作原理,不是本课程的内容。对于非计算机专业的本科生来讲,主要了解它工作的大致过程——上述五部分之间如何有机的联系在一起。

首先将自己需要计算的题目,按程序设计语言要求编写出源程序,穿好卡片、纸带或录入软盘,放在输入设备上。通过控制器由操作人员打入的命令,借助机器内部操作系统软件的支持,使输入设备等开始工作,将源程序及数据输入到存贮器,而源程序指令传输到控制器,控制器就按源程序指令的先后次序,对运算器和存贮器施发命令使其工作——运算和存取数据等。待运算结束后,将计算结果又存入到存贮器。当控制器按源程序要求把计算结果输出到外部时,则输出设备(如打印机)从存贮器相应存贮单元中取出计算结果,并通过输出设备打印或终端显示出计算结果。

§1—3 计算机语言

自学部分

信息是人们交流思想的主要工具,如语言、文字、手势、表情等等。总之人与人之间各种思想交流总是通过这些信息来传达的。人们利用电子计算机解决各种实际问题,在人和机

器之间也必须有信息的交流。当然你可以想象用人与人之间的对话方式来支配计算机，但是，目前尚未发展到完善的实用阶段。当前人们是通过“计算机语言”来实现人对机器的支配。

计算机语言在不断发展，但大体上可分为三大类：机器语言、汇编语言和高级程序设计语言。

一、 机器语言

每台机器都必须配有一套专有的用指令代码编写的语言，这种语言称为“机器语言”，用通俗的说法，机器语言就是指令代码语言。这种语言不具有通用性，每台计算机只懂得它自己这种型号机器的机器语言。

用机器语言编制的计算程序（现已淘汰）是能被计算机直接执行的，中间不需要任何“编译程序”——“翻译”。例如计算 $A = B \times C + D$ ，把这个数学表达式交给机器它是不懂的，必须换成机器语言。用DJ S-21国产机的机器语言写出的程序如下：

操作码	地址码	含义
0 0 2	1 0 1 1	取B
0 1 2	1 0 1 3	乘C
0 0 8	2 3 1 7	加D
0 0 4	1 1 5 9	送入A中

显然，这种数字代码是令人费解的，是难掌握的。这里1159，1011，1013，2317分别代表变量A，B，C，D在该机中存储单元的地址。而002，012，008，004又分别代表“取，乘，加，送入”等运算操作。由此看来用这种语言编写计算程序是一种非常消耗精力、时间的事情，只有经过专门训练的人方能掌握。既使你掌握了这台机器的机器语言，你也只能在同类型号的机器上使用，换种型号的机器语言又不一样。同时，对于不熟悉这种型号机器的机器语言的人来说，也看不懂所编程序的含义。但是它也有优点，即节省时间（因没有编译阶段），执行效率高，计算质量高等。然而终因它不易普及，难于掌握，阻碍了计算机的应用与推广。人们通常又把它叫做低级计算机语言。

需要指出的一点是，这种机器语言目前仍在机器内部使用，即使是相当先进的计算机也是这样。这是因为它所独具的那些优点所致。

二、汇编语言

它的产生是为了简化程序设计中存储分配等工作以及较为便于人们理解与掌握而设计的一种接近于机器语言的语言。它是用文字或符号表示指令的地址、操作等含义。例如上例用汇编语言可写成：

$\leftarrow B$	取B
$\times C$	乘C
$+ D$	加D
$\rightarrow A$	存入A

这种语言比较形象，便于记忆。但与普通的数学语言相比，相差还远。用这种语言写出的程序计算机不能直接执行，机器内部要有将其翻译成机器语言的软件——汇编编译。也就是说通过机器内部的汇编编译将用汇编语言编写的程序译成机器语言，只有此时计算机方可执行。

虽然这种语言比机器语言有了进步，但仍存在1)通用性差（每种型号的机器有自己的汇编语言）；2)与数学语言尚有差距，不易看懂；3)编程效率低等缺点。因此又发展了高级程序设计语言。这就是五十年代中期发展的FORTRAN以及陆续出现的ALGOL, COBOL, BASIC, PL/I, PASCAL, C, ADA等等高级程序设计语言。

三、高级程序设计语言

由于这种语言是面向用户的语言，因此它的出现，使普及、推广应用计算机成为可能。它与数学语言极为接近，并且不受机器型号的限制，因而通用性强，大大提高了用户的工作效率。有人比较过用FORTRAN语言编程序和利用机器语言编程序的工作量之比为1：(5~10)，这还不包括语言学习掌握的难易，程序的调试与修改的难易，以及在不同机器上的移植难易等等。使用高级程序设计语言编写的源程序，计算机需要通过机器内部的该种高级程序设计语言的“编译程序”翻译成机器语言后方可执行。

用FORTRAN语言将上例可写成：

$$A = B * C + D$$

其含义是将 $B \times C + D$ 的结果赋给变量A。

类似于FORTRAN语言的高级程序设计语言目前已有上千种。美国许多高等院校多采用适于教学的功能更强的PASCAL语言。下表中列出了四种主要的高级程序设计语言特性比较，每项比较按5级评分，分数越高，表明该种语言在该项中功能越好。

项目 \ 语种 \ 评分	BASIC	COBOL	FORTRAN	PASCAL
学习容易程度	5	2	4	4
编大程序适合程度	2	4	4	5
编小程序适合程度	5	2	4	4
可移植性	2	5	4	5
设计的实现性能	2	3	3	5
维修的容易程度	2	3	3	5
解决问题的功能	2	4	4	5
产生高效率的编码功能	3	4	4	4

标准化的兼容程度	3	5	5	2
总 分	26	32	35	39
平 均	2.89	3.56	3.89	4.33

由此表看出PASCAL语言是一种优点较多的语言。目前我院许多机器可以使用PASCAL语言。

§1—4 FORTRAN 语言发展概况

自学部分

FORTRAN是Formula translator 的字头缩写，意思为“公式翻译”。

FORTRAN语言从1954年发表至今已有30余年，但它经历了几个发展阶段。它最早称为FORTRAN—I，随后于1956年在美国的IBM—704计算机上使用，1958年在I的基础上作重大扩充，如引入子程序等，此时称为FORTRAN—I。以后又陆续出现了II，N…。特别值得指出的是60年代初美国开始对FORTRAN语言进行标准化。1966年制定了两级美国标准，即：

标准FORTRAN（大致相当FORTRAN—N），亦称全集。[USA. Standard FORTRAN, USAS, X3.9—1966]

标准基本FORTRAN（大致相当FORTRAN—I），亦称子集。[USA. Standard Basic FORTRAN, USAS, X3.10—1966]

1972年国际标准化组织（ISO）又为它制定为推荐标准。这个标准分为三级，即完全的；中间的；基本的（亦称一、二、三级）。

ISO 标准中

一级（完全的）相当于美国标准N。

三级（基本的）相当于美国标准I。

美国最新的标准是1978年4月3日通过的FORTRAN—77。据资料介绍美国在通过FORTRAN—77的同时，FORTRAN标准化的X3J3委员会已着手下一次的修改工作了。

本教材将以FORTRAN—N为基本内容，辅以介绍FORTRAN—77与FORTRAN—I的区别。

高级程序设计语言为什么要标准化呢？首先是因为便于用户编写程序有标准所依循，并使程序通用性强。其次是因为人们依据标准可以阅读已有程序。所以学习高级程序设计语言既要注意与机器对话，又要注意与人对话。

虽然FORTRAN语言已标准化，但ISO组织是松散组织。计算机厂家生产的计算机均有该机器的FORTRAN“语言文本”或是“用户手册”。所以上机前，应熟悉所使用的机器对FORTRAN语言是否有特殊规定。

§1—5 用高级程序设计语言解题的全过程

讲解部分

用高级程序设计语言解题的过程与步骤，如图1—2所示

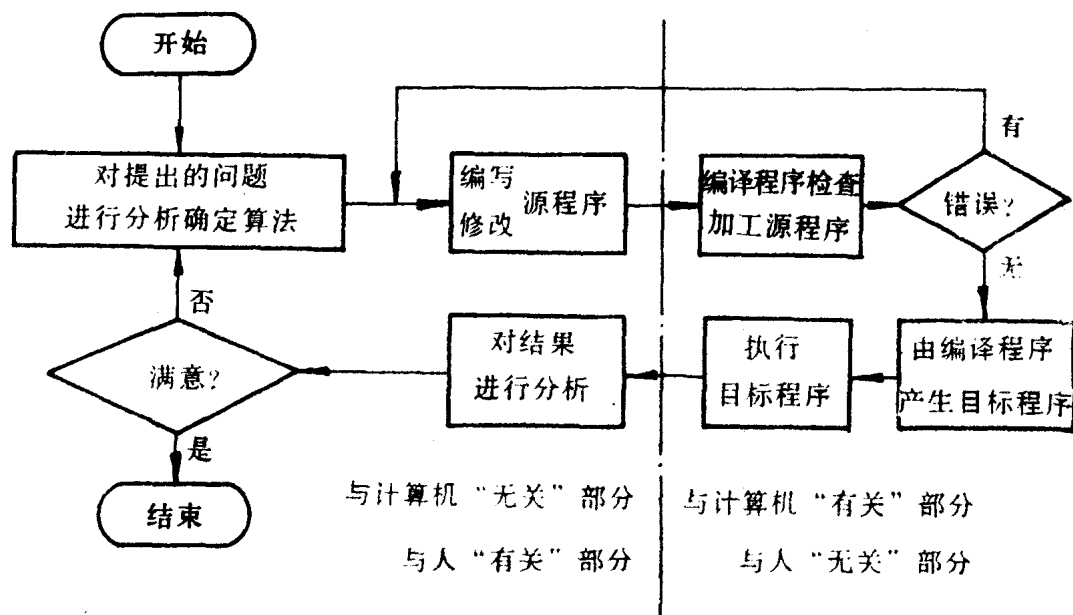


图 1—2 解题过程与步骤

所谓“源程序”，就是人们用高级语言所编写的程序。而“编译程序”是系统软件一部分，事先已在计算机内部安装。它的作用是把源程序按所使用的语言（各种语言有各自的编译程序）规范进行检查，并将源程序“翻译”成机器能执行的“目标程序”。源程序通常在输出计算结果前打印出来。如有错误，它可按规定打出错误信息，指出错处出处、错误的原因（通常还需要根据打印的错误信息翻阅该机器的错误信息表）。而目标程序通常是不打印出来。即使打印出来，也只有经过专门训练的懂得该机器的机器语言的人才能读懂。

（图 1—3 示出了机器中的错误检查。）

显然要使自己的计算获得满意的结果，源程序的编写是个十分重要的工作。而它正是本课程教学的最终目的。

概括起来用计算机解题可分为三个阶段

第一阶段——编程阶段。

第二阶段——编译阶段。在该阶段主要由编译程序完成对源程序的编译，其中包括检查源程序有无语法错误；把没有语法错误的源程序加工为

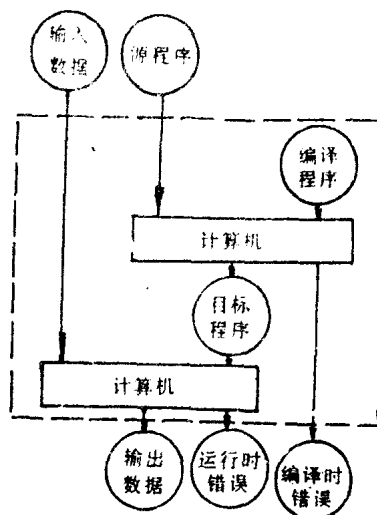


图 1—3 机器中的错误检查

机器能够理解与执行的目标程序（注意目标程序通常为机器语言）。

第三阶段——执行阶段。在该阶段还可能出现由于源程序中有逻辑错误；数据准备不

当：影响正常运行的计算错误如负值开平方等等而引起运行错误。（计算机对运行中错误也将在输出设备上显示出错误信息。）当源程序运行中没有出错时，则将有计算机结果在输出设备上显示出来。

§1—6 FORTRAN程序举例

讲解部分

高级程序设计语言犹如一门外语。如果一开始就逐条讲述它的语法规则，会使初学者感到不得要领，枯燥无味。为此先举一例，使初学者能够对源程序有一概貌了解，对FORTRAN语言有些感性认识。对于此例初学者只要了解大意即可，不必深究细节，因为以后要一一详介。

例一、

一、题 目

计算负温度系数热敏电阻 $R_t = R_{20} e^{B(\frac{1}{273+t} - \frac{1}{293})}$ 与固定电阻 R_1 并联后的等效电阻 R 在 $(20^\circ\text{C} \sim 100^\circ\text{C})$ 范围内，每隔 10°C 的阻值。

其中已知： $R_{20} = 433 \Omega$ ；

$R_1 = 620 \Omega$ ；

$B = 2800$ （物理常数）

二、分 析

根据电路知识，等效电阻

$$R = \frac{R_1 \cdot R_t}{R_1 + R_t} = \frac{R_1 \cdot R_{20} \cdot e^{B(\frac{1}{273+t} - \frac{1}{293})}}{R_1 + R_{20} \cdot e^{B(\frac{1}{273+t} - \frac{1}{293})}}$$

显然为了计算 20°C ， 30°C …… 100°C 时等效电阻值，只要将已知的 R_1 ， R_{20} ， B 代入上式即可，且要进行 9 次计算步骤一样的重复运算。

三、画流程图（框图）

流程图就是按统一规定的一些几何图形，用流线连接起来，能反映出计算机执行顺序的图。它可以帮助人们编程也可以帮助人们阅读和理解已有程序，交流程序设计思想。

下表列出了流程图规定的几何形状和含义。

流程图符号及含义