胡金柱 主编

FORTRAN 77
与应用软件开发方法

化学工业出版社

FORTRAN77与应用软件开发方法

胡金柱　主编

化学工业出版社

内 容 摘 要

本书用软件工程学的观点介绍了开发应用软件的一些具体而实用的方法；详述了FORTRAN77语言以及用它进行结构化程序设计的基础方法和风格；探讨了FORTRAN77与BASIC、FORTRAN77与C-dBASEⅢ进行数据交流、相互调用的方法。另外，还介绍了软件开发中的汉字处理技术。

本书的特点是深入浅出，通俗易懂，内容新颖，具体实用；既注重了科学性和系统性，又注重了实用性；既注重了广度，又注重了深度，重点内容作了理论与方法上的论述。

本书适用面广，可供管理、应用计算机的人员学习参考，也可以作为大专院校、中等专业学校学生学习计算机软件的教材和教师的教学参考书。

FORTRAN77与应用软件开发方法

胡金钰，主编

责任编辑：刘 哲
封面设计：许 立

*

化学工业出版社出版发行
（北京和平里七区十六号楼）
化学工业出版社印刷厂印刷
新华书店北京发行所经销

*

开本787×1092 1/16 印张 19 1/4 字数480千字
1991年7月第1版 1991年7月北京第1次印刷
印 数 1—2,300
ISBN 7-5025-0654-3/TP·17
定 价13.00元

前　言

在多年的科研与教学实践中，我们体会到，有关软件工程方面的著作都是高层次的，不能满足一般读者的需要，不能指导他们结合自己的工作和专业开发实用软件。而许多计算机语言方面的著作往往又是纯普及性的，不利于读者在普及的基础上进一步得到提高。将这两者有机地结合在一起，以满足广大读者的需要，是我们多年的愿望和努力方向。后经专家们的指导，与同行多次磋商，并参考了大量的中外文献和著作，终于编写了此书。

本书共分十五章。其中二至八章介绍了FORTRAN77语言及其程序设计方法，第八章还介绍了如何用FORTRAN77与BASIC进行数据交流、相互调用的方法；第九章介绍了软件开发的计划工作，重点介绍了经费估算、进度安排和计划说明书的编写等方法；第十章和第十一章重点介绍了结构化分析与设计方法，并以设计模糊聚类分析软件为例，系统介绍了应用软件的分析设计过程；第十二章介绍了程序的详细设计概念、结构程序设计基础与风格，分析了程序的三种控制结构及其设计表示法；第十三章介绍了软件测试方法；第十四章介绍了软件开发中的汉字处理技术，应用软件开发中数据库设计的一般方法，以及FORTRAN77与C - dBASEⅢ进行数据交流的方法。

本书的编写出版得到了软件工程学专家何克清副教授的大力支持和帮助，他在百忙中抽时间认真审阅了全稿，在此向他表示衷心地感谢。本书 1～13章由胡金柱同志负责编写，14章由常继康同志负责编写。曹琦、彭琳、王林平、魏开平等同志参加了编写、腾抄和校对工作。编写过程中还得到了许多同事的支持和帮助，在此一并致谢。

作　者

目　录

第1章 概 述

1.1 程序设计语言的基本概念

电子计算机（亦称电脑）是一种能够自动地、高速地用于计算和信息处理的现代化电子设备，是替代人类脑力劳动的有效工具。无论电子计算机解决什么问题，都离不开编写程序。所谓程序，就是根据具体问题，用计算机语言编写的解题步骤。编写解题步骤的过程即为程序设计，编写程序的语言即为程序设计语言。机器语言、汇编语言以及高级语言都是计算机程序设计语言。

机器语言与汇编语言都是面向计算机而不是面向一般用户的计算机语言，所以人们把它们统称为“低级语言”。

高级语言是面向用户而与计算机的内部结构无关的语言。用高级语言编写程序，可以不考虑计算机的内部逻辑结构，而只注重解题的方法和实现的过程。所以，高级语言具有简单易学，使用方便等优点。

用高级语言编写的程序称之为“源程序”。计算机不能直接执行这些源程序，必须转换成用机器指令表示的“目的程序”才能执行。

把用高级语言编写的“源程序”转换成“目的程序”的工作，是由“编译程序”自动完成的。编译程序加上一些辅助程序（如连接程序）统称为“编译系统”。图1.1是一个由源程序到计算结果的编译系统工作过程示意图。

编译系统对源程序进行编译的同时，还要自动检查源程序中的错误，并打印出“出错信息”。程序设计者根据这些错误信息去修改源程序，然后重新编译。编译成功便得到目的程序，再执行连接程序。连接成功之后，便得到“可执行目的程序”。最后，由计算机执行目的程序，并输出计算结果。

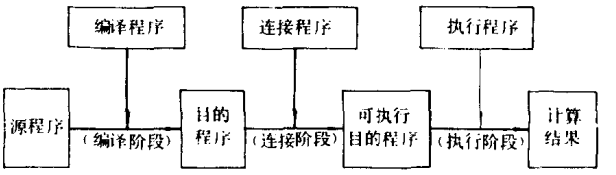

图 1.1

1.2 FORTRAN77语言简述

FORTRAN77语言是目前国际上广泛流行的计算机高级程序设计语言之一。FORTRAN是英文FORmula TRANslation的缩写，其含意是“公式翻译”。

FORTRAN语言是世界计算机史上的第一个高级语言，它最早出现于1956年，后经多次修改与完善，于1978年4月美国国家标准协会（Americal National Standards Institute，简称ANSI）正式发表了新的FORTRAN文本（即ANSIX3.9-1978），叫做FORTRAN77。美国的FORTRAN77文本一发表，很快就被国际标准化组织（ISO）采用，并作为国际标准FORTRAN文本。我国国家标准总局于1982年颁布的《程序设计语言FORTRAN》标准也采纳了FORTRAN77。

FORTRAN77与其它计算机语言相比，具有标准化程度高、通用性强、使用灵活方便等特点。FORTRAN77的另一个特点是具有较强的结构程序设计能力和字符处理功能，所以它特别适用于处理大而复杂的科学计算问题，是目前用于科学计算和数据处理的最理想的高级语言之一。

FORTRAN77的源程序由一系列的语句组成。FORTRAN77的语句可以分为五种基本类型：（1）向存贮器存取数据；（2）执行计算操作；（3）输入、输出；（4）控制程序的执行；（5）说明语句。其中前四类是可执行语句，是可以由编译程序翻译成对应机器指令的语句；而说明语句是用于说明程序的内容和属性，为可执行语句作好必要的准备的语句。所以，一般情况下，源程序中的说明语句应放在可执行语句之前。一个可执行的源程序必须有可执行语句，但可以没有说明语句。

例1.1 求N个（N为任意自然数）变量x的平均数。其程序可以是：

```
      PROGRAM EXA1
      SUM=0
      WRITE(*,*)'Enter count N:'
      READ(*,*)N
      DO 15 I=1,N
        WRITE(*,*)'Enter the value of X:'
        READ(*,*)X
        SUM=SUM+X
15    CONTINUE
      IF(N.NE.0)THEN
        AVERAG=SUM/N
      ELSE
        AVERAG=0
      ENDIF
      WRITE(*,*)'The summin is:',SUM
      WRITE(*,*)'The count is:',N
      WRITE(*,*)'The average is:',AVERAG
      END
```

以上程序的第一行是给保存求和总数的变量SUM清零；第二行是输出一行提示，输入要求平均数的变量x的个数N；第三行是输入语句，要求从终端键盘上输入N的值；第四行是DO循环控制语句，表示五～七行重复执行N次，这三行叫做“循环体”，作用是输入N个x的值，并求N个x的总和；第八行是与第四行配合构成循环结构的终端语句，前面的15是该语句的语句标号；第九行～十三行是一个IF控制块，作用是判断N是否为0，若不为0，则用N除以SUM，便得N个x的平均值，但若$N=0$，则置平均数为0；第十四～十六行用于输出结果，包括N的个数、N个x值的和、平均数；第十七行是程序结束标志语句。每个FORTRAN77程序的最后都要有一个END语句。END语句也是控制语句，当程序执行到END语句便控制计算机停止执行这个程序。

以上程序的算法还可以用图1.2表示。图1.2叫做“流程图”(Flow Chart)，或者叫“程序框图”，常简称“框图”，缩记“FC”。图1.3列出了最常用的流程图符号及其符号名称。

1.3 源程序的输入与书写规则

输入FORTRAN77源程序时，必须严格按照规则进行操作，否则编译程序无法识别，编译过程无法通过。当编译程序查出源程序中的语法错误后，也必须按规则进行修改。编写程序时要按规则书写，书写源程序还可以使用程序纸。

1．语句标号的书写

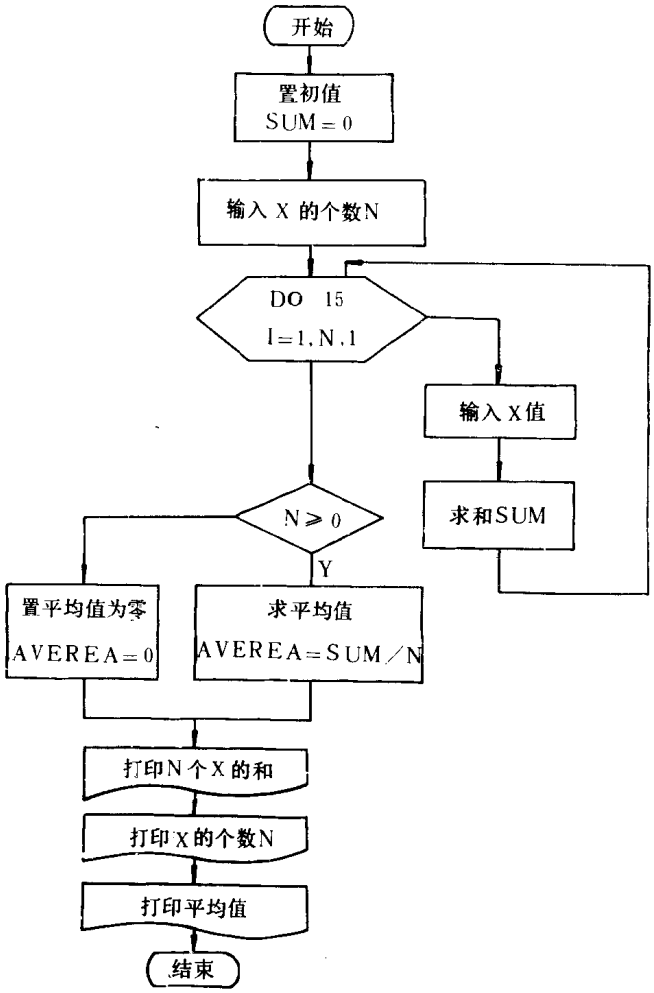

图 1.2

序号	框图名称	框图符号
1	开始框、结束框	
2	处理框	
3	输出框	
4	判断框	
5	循环框	
6	流程框	

图 1.3

语句标号只能是1~5位的十进制正整数，所以语句标号的范围是 1 ～99999。语句标号左边的零无意义，例如185与00185是同一个语句标号。语句标号只起标识作用，本身无数值的大小区别，也无任何次序上的要求。

语句标号必须写在一行的第 1 列到第 5 列字符位置上，这个位置叫做“标号位”。

在同一个程序中，不同的语句不允许有相同的语句标号。

3．语句行的书写格式

一般情况下，一行只能写一条语句。但如果一条语句很长，在一行内写不完时，可以接在下一行继续写。这条语句的第一行叫做“起始行”，以后各行叫做“继续行”。语句行的书写规则是：

（1）起始行：第 1 ～ 5 列为空格或语句标号；第 6 列必须是零或空格；第 7 列～72列为语句正文。

（2）继续行：第 1 ～ 5 列为空格；第 6 列必须是非零或者非空格字符；第 7 ～72列为续行正文。续行最多可以有18行。

（3）语句正文可以写在第 7 ～72列的任意位置。

4．注释行

注释行的功能是说明整个源程序或者程序中某个部分的内容和作用，使其更加清晰，增加程序的“易读性”。在注释行内可以包括计算机系统能识别的任何字符。编译程序在编译源程序时对注释行的内容不翻译，只在列源程序清单时，照原样印出其内容。所以注释行可以出现在源程序的任何位置，且毫不影响程序的运行。

注释行的书写规则是：

（1）在第一行的第 1 列上写上一个星号（*），或者一个大写字母C，该行就是注释行。星号（*）和大写字母C称为“注释标识符”。注释内容写在第 2 列以后的任何位置。

（2）空白行也是注释行。

（3）注释行不存在继续行问题。如果某个注释内容在一行内写不完，可以在下一行接着写下去，但是每一行的第 1 列都要加注释标识符 * 或者C。

5．结束行（END语句）

一个程序单位的最后一行叫做“结束行”。结束行必须是END语句。一个结束行就是在 7 ～72列之间的任意位置上写上END三个字母（它们之间可以插入任意个空格）。

END语句表示本程序单位内的所有语句行和注释行到此结束。结束行后面的语句均不属于本程序单位。

6．主程序开始语句行

FORTRAN77语言的源程序包括主程序和子程序两大类。可执行的源程序必须有一个主程序而且只能有一个主程序，可以同时包含一个或多个子程序，也可以没有子程序。任何主程序都可以用以下形式的语句开头：

PROGRAM PM

其中，PROGRAM　是主程序开始语句的“关键字”（或叫“定义符”）

PM　　是给主程序取的名字，叫做“主程序名”。

主程序的开始语句行可以省略不写。

1.4 计算机系统

电子计算机系统由硬件系统和软件系统两大部分组成。只有硬件系统和软件系统相互配合，才能充分发挥电子计算机的作用。

1．计算机的硬件系统

计算机硬件是指计算机系统的各种实际装置和物理设备。图1.4是一个典型的计算机硬件系统的示意图。一个计算机硬件系统主要由中央处理机、存贮器、输入和输出设备组成。

输入设备是人-机联系的主要设备。其主要作用是将人们编写的程序和要处理的数据输入给计算机的主存贮器。常用的有终端输入机（终端键盘或者电传）、卡片读入机、磁带、磁盘等。

存贮器包括主存贮器（俗称“内存”）和辅助存贮器（俗称“外存”）。内存的主要作用是用来存放当前正在执行的程序和原始数据，暂时存放计算的中间结果和最后结果。内存的特点是存取信息的速度快，而且只存放当前正在执行的程序和数据等，暂时不用的程序存放在外存。外存比内存的存贮容量大，是内存的补充，但存取信息的速度比内存慢。外存不能直接和CPU交换信息，一般只能和内存成批交换信息。常用的外存有软磁盘、硬磁盘、磁带等。

存贮器被分成许多小的存贮单元。一般情况下，每个存贮单元包含若干个二进制位，每个二进制位称为一个“比特”(bit)。目前的计算机多以八个二进制位组成一个字节（byte）。一个存贮单元由一个或者多个字节组成，称为一个“字”（word）。每个存贮单元中存放一个数据或一条指令。每个存贮单元都有一个编号，称为“地址”。现代计算机存贮器的存贮容量一般以K和M为计量单位：1 K＝1024个字节，1 M（称作“兆”）＝1024 K＝1024×1024个字节。

中央处理机是计算机的核心，它由运算器和控制器组成，其作用是控制计算机的各个部分按照程序的要求协调一致地工作，执行各种算术运算和逻辑运算，控制输入、输出等工作。

常用的输出设备有：终端显示器（CRT）、各种打印机、绘图仪和磁盘、磁带机等。用户要查看其输入到计算机内的程序、数据以及程序执行的结果，都可以通过输出设备而得到。

中央处理机和内存为计算机硬件系统的“主机”部分，输入、输出设备和外存为计算机硬件系统的“外围”部分。

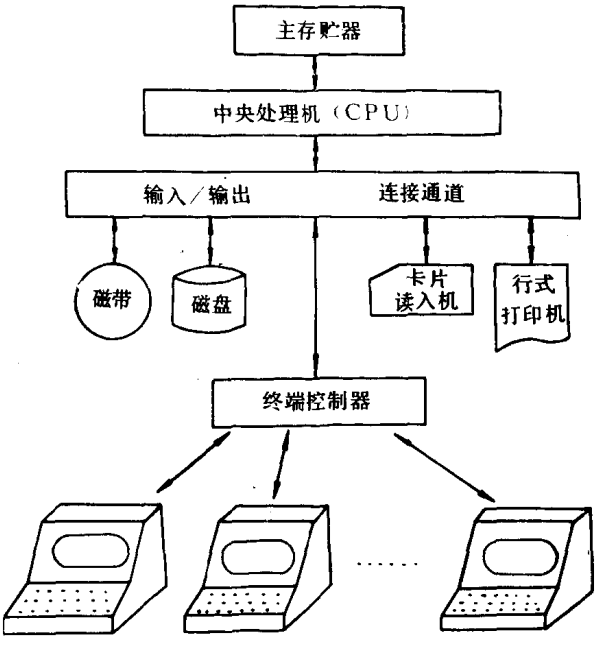

图 1.4

2．计算机的软件系统

计算机软件是一些抽象的、逻辑性的产品，其核心之一是“程序”。软件不是物理实体，它很容易被修改，只要更改程序中的几条语句，就有可能使软件系统的功能发生根本性的变化。所以，“易被修改性”是软件系统的重要特征之一。

软件又可以分成两大类，即系统软件和应用软件。系统软件是用于计算机系统(软件和硬件)本身的管理、维护、控制和运行，以及对计算机应用程序进行翻译、装入、管理、维护、控制和执行的软件。当前最典型的系统软件是计算机操作系统(Operating System)，操作系统的主要功能是扩展计算机的功能，方便用户使用计算机，提高计算机的使用效率。应用软件是为用户解决某一个特定问题而设计的软件，是在系统软件的基础上开发的、需要系统软件支持才能运行的、面向用户的软件。所以，系统软件是一个计算机系统的基础软件，而应用软件是计算机系统的目标软件。系统软件可以为所有的计算机用户服务，而应用软件只能为特定问题的用户服务。应用软件是计算机普及应用的直接者，是计算机系统与用户的接口。

1.5 计算机软件发展简述

计算机软件的发展是与计算机硬件技术的发展、计算机的推广应用紧密相关的。软件的发展一般分成三个阶段。

早期的计算机上几乎没有什么软件，用户直接用机器指令或者汇编语言编写程序。这种程序的设计十分麻烦，编制和调试一个稍大一点的程序，常常需要一年以上的时间。而且，计算机程序的设计、调试、使用和维护等工作只有少数专家才能完成，程序设计只是他们头脑中隐含的过程和"艺术"。这种局面严重地影响了计算机的普及应用。

1956年，FORTRAN语言的出现，标志了第二阶段的开始。运用高级语言编写程序简化了程序设计过程，程序设计工作不再是少数专家才能做到的事。一般工程技术人员经过短期培训，就可以使用计算机解题了。这样就大大地缩短了解题周期，扩大了计算机的应用领域，推动了计算机的发展。

六十年代以来，由于电子技术、高级语言与编程技术的发展，计算机的应用日益渗透到国民经济、国防建设和科学文化事业的各个领域，涉及到社会生活的各个方面，如企业管理、人事档案管理、商业事务处理、飞机订票等等。这些应用系统的软件一般都比较大，逻辑比较复杂，而且需要不断地更改和扩充其功能。这些大系统常常需要花费大量的物力、人力和时间，而研制出来的产品却可靠性差、错误多、维护和修改也很困难，有些甚至彻底失败了。有些系统虽然完成了，但比原计划推迟了几年，而且经费大大超过了预算。有些系统未能达到用户当初的要求。有些系统无法进行修改和维护。不仅如此，由于软件规模的增大，人工费用的迅速增涨，使得软件生产的费用急剧上升。另一方面，由于集成电路的出现和发展，计算机硬件的功能和质量得到了不断地提高，其价格大幅度地下降，因此，软件与硬件的投资百分比逐年变化，而且差距越来越大。以美国为例，1955年软件生产投资只占20%以下，而1970年已上升到60%，1985年已达到90%左右(如图1.5所示)。这些就是所谓"软件危机"的表现。

"软件危机"说明了原有的软件生产技术跟不上需求的发展(如图1.6所示)，这个矛盾随着软件复杂程度的增加更加突出，这就促使人们去考虑解决现实矛盾的办法。在实践中，人们逐步认识到：用编写小程序的"作坊式"的手工艺方式来研制软件已经行不通了，必须寻找新的方法来指导软件的研制。在研究软件生产的新技术、新方法过程中，产生了新的学科——软件工程学。软件工程学的出现标志着软件发展的第三阶段的开始。"工程化"的思想为软件的开发指明了新的道路。目前，国内外越来越多的人在用工程化的思想与方法开发新的软件。

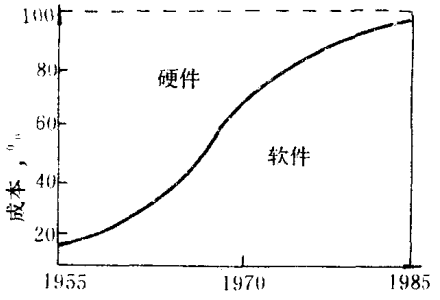

图 1.5 计算机系统硬、软件成本比例的变化

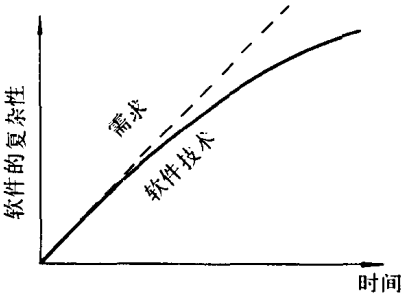

图 1.6 软件技术的发展落后于需求，差距日益增大

1.6 软件生存周期

软件生存周期（Software Cycle）是指从提出某个软件项目开始，到开发成功并投入运行，直到该软件报废或停止使用为止的全过程。软件生存周期一般划分为软件计划、软件需求分析、软件设计、编码、软件测试、软件运行与维护等六个阶段。

（1）软件计划 主要目的是了解并确认将要开发的软件“做什么”，是否可以实现。主要任务是通过调研确定软件项目的范围，研究开发该软件项目的经济、技术上的可行性，并预算人力和物力资源，初步确定开发成本和进度。

（2）需求分析 主要目的是确定目标系统必须具备的功能、作用。主要任务是进一步认识并解决用户提出的“做什么”的问题。在此阶段，要对用户的要求作具体地分析研究，并用“软件需求规格说明书”（又叫“系统说明书”）表达出来，作为用户和软件开发人员之间的共同约定。

（3）软件设计 主要目的是解决“如何做”的问题。设计阶段又可以分为总体设计和详细设计。总体设计的主要任务是选择设计方法、确定软件系统的模块结构，确定模块之间的相互联系，设计并描述每个模块的功能。详细设计则是设计实现每个模块功能的具体算法。

（4）编码 主要目的是要得到实现每个模块功能的具体程序。所以编码阶段的任务是选择适当的程序设计语言，按照软件设计说明书的要求为每个模块编写具体的程序，并对每个程序作些初步的检测和调试。

（5）测试 主要目的是通过各种类型的测试和调试，得到可以运行的软件，并使软件达到预定的要求。主要任务是要发现并排除程序中的错误，使之能够安全可靠地运行。

（6）运行与维护 主要目的是通过各种必要的维护活动使软件系统能够始终满足用户的需要。

在以上软件生存周期中，前五个阶段称之为“软件的开发期”，最后一个阶段称之为“软件的运行维护期”。

为了保证软件产品的质量和使用维护的方便，软件工程学对软件开发期的每个阶段提出了如下要求：

（1）必须有一个整体规划（即计划书）；

（2）必须有一种协调软件开发的各个方面、各个层次的通讯手段；

（3）必须有一本帮助记忆的工作记录。

以上要求的实现，就是软件开发过程的技术“文档”（Document）的产生。

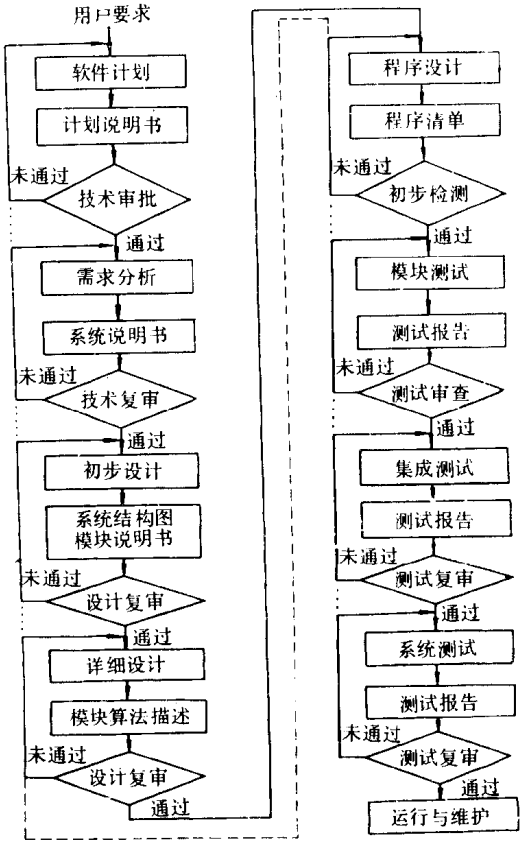

图 1.7

软件开发的每个阶段都必须产生一定规格的文档，这些文档的集合就是软件。从开放期的需求分析阶段开始，每个阶段的工作都必须在上一阶段提供的文档基础上进行。在每个阶段结束之前，管理人员都必须从技术和管理两个方面进行严格审查，合格之后才能交给下一阶段工作。所以，软件开发不是简单的程序设计，而是计划、分析、设计、编程以及测试、审查和验收的全过程。

但是，实际应用软件的开发过程不可能是直线进行的，经常存在着反复。当某一个阶段的工作通不过，软件开发人员就必须返回到前面，查找存在的问题及其原因，有时还需彻底推翻前阶段的工作，重新开始。图1.7描述了软件开发期各阶段之间的联系，图中虚线表示可能需要返回到更上一个阶段才能解决问题。

1.7 软件开发的目标

用软件工程方法开发应用软件的目标是：以较低的投资而获得高质量的软件。评价软件质量高低的主要指标是软件的可靠性、易维护性和高效性。

1．软件的可靠性（Software Reliability）

软件的可靠性是指在规定的条件下和预期的时间内，能正确地完成所要求的功能任务。一个可靠的软件，必须是正确无误的、完整无缺的、安全可靠和保密性强的软件。

一个安全可靠的软件系统还应具备抵御各种错误信息的能力，应具有自动检测的功能。在遇到某种意外的情况时，能按某种事先规定好的方式作出适当地处理，并能有效地控制事故的蔓延，从而避免严重损失。例如，用户有意无意地输入了一些错误信息，软件应该有能力对那些明显会造成除法异常、不符合物理性质或几何性能的数据进行检测与处理；突然停电或者计算机硬件出现某个故障时能及时保存重要信息，或者通知管理人员。

综上所述，软件“可靠性”的含义包括两个方面：

（1）正确性　软件系统本身没有错误，在正常情况下能够正确地工作。

（2）健壮性　软件系统在意外的情况下仍然能够适当地工作，而不会造成意料不到的严重损失。

2．软件的易维护性（Software Maintainability）

软件的维护包括两重含义：（1）用户在使用一个软件时可能要根据具体情况对该软件进行适当的修改，以满足用户在新条件和新技术情况下的要求；（2）软件经测试后还可能含有错误，这些错误在它投入运行时会逐步暴露出来，所以运行阶段还需要继续排错。因此，一个软件系统在运行阶段还要不断地修改或扩充，这就叫做“软件的维护”。

软件维护的工作量相当大，也相当困难。许多统计资料说明，软件维护工作的费用占整

个软件投资的40～60%，维护工作量占整个软件生存周期的60～70%。而软件维护的难易程度又是与软件的易维护性密切相关的。软件的易维护性越好，软件维护的难度越低。

软件的易维护性，是指软件容易被人阅读和理解，容易测试，容易修改。因此，影响软件易维护性的主要因素有：

（1）易理解性　一个软件不仅要给计算机执行，还要经常供人阅读，例如审查各阶段的工作、测试、排错和修改都需要阅读。所以，易维护性好的软件必须是容易被人读懂并理解的软件。

“易理解性”有两个含义：一是指软件的功能和内部结构清晰，容易被软件开发人员阅读和理解；二是指软件的人-机接口简单明了，容易被用户接受和使用。

（2）易测试性　易测试性是指证实软件正确性的难易程度。一个容易测试的软件首先应该是容易理解和可靠的，其次是软件中程序的简明性，程序越复杂越不容易测试。

（3）易修改性　易修改性是指改变软件的难易程度。一个易修改的软件应该是易理解的、通用的、灵活的和简明的。其中，通用性是指不需要作很多的修改就可以使软件改变它的功能作用；灵活性是指软件容易被分解或组合。

为了增加软件的易维护性，在软件开发的每一阶段都要考虑尽可能地为用户提供一个易于维护的软件。在完成每个阶段的工作之后，都要从软件的易理解性、易测试性、易修改性等三个方面进行易维护性复审。

3．软件的效率（Software Efficiency）

效率是指软件系统是否能够有效地使用计算机的各种资源，如CPU时间和存贮空间等。所以，解决同一个问题的程序，如果程序中的语句和占用计算机的存贮空间最少，就认为是效率最高。以前，由于硬件价格昂贵，为了提高效率，人们千方百计地追求所谓的“程序设计技巧”。但是，技巧性越高的软件，越是难于阅读，难于理解。这说明追求效率与追求易维护性、可靠性往往是相互矛盾的。一般情况下，软件的高可靠性需要以一定的时间和空间作为代价。所以片面强调节省时间和空间，设计出来的软件就可能结构比较复杂，难于测试和修改，难于维护，难于保证软件的可靠性。

七十年代以来，计算机的运行速度和存贮容量成倍增加，硬件价格下降，人工费用直线上升，于是程序的执行速度和占用存贮空间的多少已不是主要矛盾了。今天，人们普遍认为，一般情况下宁可降低一些计算机的硬件效率，也要保证软件具有高可靠性，结构层次清晰，易于阅读和理解，易于测试和维护。

可靠性、易维护性和高效性等三个评判软件质量的指标之间既互相联系，又互相矛盾。另外，各指标的重要程度又随着一个软件所处的工作环境和作用范围的不同而不同。因此，在开发一个应用软件时，应根据具体的条件和情况，充分考虑各种可能的候选方案，在各种矛盾的指标之间权衡利弊，以最低的代价，最大限度地获得高质量的软件，达到软件开发的最终目标。

第2章 FORTRAN77基础

FORTRAN 77的源程序由一系列的语句组成。一般情况下，一条语句包括关键字（或叫“语句定义符”）和语句“实体”（即语句内容），而组成语句的最基本成分是字符。常数、变量、内部函数和数组是参加程序运算的基本数据结构，这些数据叫做“运算对象’或者“操作数”。FORTRAN 77允许使用整型、实型、双精度型、复型、逻辑型和字符型等六种类型的操作数，用运算符把各种类型的操作数连接起来便组成了表达式。

2.1 FORTRAN77的字符集

计算机程序设计语言也和英语等各种自然语言一样，规定了它允许使用的基本字符。FORTRAN 77语言的基本字符由26个大写英文字母、10个数字和13个专用字符组成。

1．通用字符

（1）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等二十六个英文字母。

（2）0、1、2、3、4、5、6、7、8、9等十个数字字符。

2．专用字符

$	美元符	+	加 号	:	冒 号
.	小数点	-	减 号	=	等 号
(	左括号	/	斜 杠		空 格
)	右括号	'	撇号(单引号)		
*	星 号	,	逗 号		

其中，空格符又叫做“空白字符”，书写时常用“␣”表示。

2.2 常数与变量

为了适应各种不同的处理对象，FORTRAN 77允许在程序中使用以下六种类型的数据：

（1）整型（INTEGER）

（2）实型（REAL）

（3）双精度型（DOUBLE PRECISION）

（4）复型（COMPLEX）

（5）逻辑型（LOGICAL）

（6）字符型（CHARACTER）

其中，整型、实型、双精度型和复型等四种数据统称为“算术型”的数据。

2.2.1 常数

常数是一些已知的、在程序执行过程中始终不变的数据。

1．整型常数

整型常数简称“整常数”或“整数”。整常数是不带小数点的十进制数字串，如3，-22，0，