

· 高等学校计算机基础教育教材精选 ·

# 程序设计基础

## —— Fortran 95

闫彩云 王红鹰 主编  
李玉龙 主审



清华大学出版社

· 高等学校计算机基础教育教材精选 ·

---

# 程序设计基础

## —— Fortran 95

---

闫彩云 王红鹰 主编

李玉龙 主审

王丽娟 段志东 张翠玲 未碧贵 方红兵 参编

清华大学出版社  
北京

## 内 容 简 介

本书全面、系统地介绍了 Fortran 95 的语法规则以及利用它进行程序设计的方法。主要内容有 Fortran 95 概述及编译环境的介绍、Fortran 95 程序设计基础、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数与子程序、文件、派生类型与结构体、指针、模块、常用数值算法。

本书针对初学者的特点,突出基础知识的讲解,全书概念清晰,语言简单易懂,实例丰富,运行结果直观可靠。可作为高校理工科类学生学习程序设计的教材,也可作为程序设计的初学者、从事工程计算的工作人员和科研人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

## 图书在版编目(CIP)数据

程序设计基础——Fortran 95/闫彩云,王红鹰主编. —北京:清华大学出版社,2011.3  
(高等学校计算机基础教育教材精选)

ISBN 978-7-302-24865-1

I. ①程… II. ①闫… ②王… III. ①FORTRAN 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 031125 号

责任编辑:焦虹 顾冰

责任校对:梁毅

责任印制:杨艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, [jsjcc@tup.tsinghua.edu.cn](mailto:jsjcc@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260

印 张:18.25

字 数:435 千字

版 次:2011 年 3 月第 1 版

印 次:2011 年 3 月第 1 次印刷

印 数:1~3000

定 价:29.00 元

---

产品编号:040838-01

# 前言

程序设计是高等学校计算机基础教育的基础与重点,目的是向学生介绍程序设计的基础知识,使学生掌握高级语言程序设计的基本思想和方法,培养学生掌握用计算机处理问题的思维方法。

FORTTRAN 语言是世界上最早出现的计算机高级程序设计语言,从 1954 年第一个 FORTRAN 语言版本的问世到现在,FORTTRAN 语言的标准化不断吸收现代化编程语言的新特性,FORTTRAN 语言就以其特有的功能在数值、科学和工程计算领域发挥着重要作用,并且在工程计算领域占有重要地位,很多优秀的工程计算软件都是运用 FORTRAN 语言编写,如 ANSYS、Marc 等。

基于 Windows 平台下的 Fortran 90 的推出,使 FORTRAN 真正实现了可视化编程,彻底告别了传统 DOS 环境(字符界面),转到了现代 Windows 环境(视窗界面),共享微软公司 Windows 平台的丰富资源。本书以 Fortran 95 为平台,介绍程序设计的思想和方法。

本书以程序设计为主线,以编程应用为驱动,通过案例和问题引入内容,重点讲解程序设计的思想和方法,内容全面,概念清晰,语言简单易懂,实用性强。

书中所有程序实例都由授课教师在多年授课过程中精挑细选所得,并采用目前流行的可视化的 Microsoft develop studio 集成开发环境,使读者在程序的思维训练和程序组织方面得到极大简化。

为使读者更好地掌握 Fortran 95 程序设计基础,我们还编写了配套的《Fortran 95 程序设计上机指导、习题答案及测试题》,可作为学习参考书。另外,还有与本书配套的电子版的教学课件,供教师教学参考使用。

本书可作为高校理工科类学生学习程序设计的教材,也可作为程序设计的初学者、从事工程计算的工作人员和科研人员的参考书。

本书由闫彩云、王红鹰主编,李玉龙主审。本书第 2~5 章由闫彩云编写,第 1、10、12 章由王红鹰编写,第 7 章 1~4 节和附录 A 由王丽娟编写,第 7 章 5、6 节和第 11 章由段志东编写,第 8 章和附录 B 由张翠玲编写,第 6 章 1~3 节和第 9 章由方红兵编写,第 6 章 4~8 节由未碧贵编写。

本书在规划、编写过程中得到了兰州交通大学教务处、数理与软件工程学院、土木工程学院、环境与市政工程学院的领导和老师们的大力支持。作者在此表示衷心的感谢。

鉴于作者水平所限,书中难免有不当或错误之处,恳请读者不吝赐教。

作 者

2010年11月

# 目录

<b>第 1 章 Fortran 95 概述</b> .....	1
1.1 程序设计与程序设计语言 .....	1
1.2 FORTRAN 语言发展简史 .....	2
1.3 Fortran 95 语言的特点 .....	3
1.4 程序设计引例 .....	4
1.5 Fortran 95 编译环境与上机步骤 .....	7
1.5.1 Compaq Visual Fortran 6.5 的安装与启动 .....	7
1.5.2 上机步骤 .....	7
习题 1 .....	17
<b>第 2 章 Fortran 95 程序设计基础</b> .....	18
2.1 Fortran 95 的字符集、标识符和关键字 .....	18
2.1.1 字符集 .....	18
2.1.2 标识符 .....	19
2.1.3 关键字 .....	19
2.2 Fortran 95 程序的书写格式 .....	19
2.2.1 固定格式 .....	20
2.2.2 自由格式 .....	20
2.3 Fortran 95 的数据类型 .....	21
2.3.1 数值型数据的表示及存储 .....	21
2.3.2 非数值型数据的表示及存储 .....	22
2.4 常量和变量 .....	23
2.4.1 常量 .....	23
2.4.2 变量 .....	28
2.5 Fortran 95 的算术运算符与算术表达式 .....	31
2.5.1 算术运算符 .....	32
2.5.2 算术表达式 .....	32
2.6 Fortran 95 标准函数 .....	33
习题 2 .....	36

<b>第 3 章 顺序结构程序设计</b>	37
3.1 赋值语句	38
3.2 输入和输出语句	40
3.2.1 表控输出输入	41
3.2.2 格式化输出输入	43
3.3 end 语句、stop 语句和 pause 语句	48
3.3.1 end 语句	48
3.3.2 stop 语句	49
3.3.3 pause 语句	49
3.4 程序举例	49
习题 3	52
<b>第 4 章 选择结构程序设计</b>	54
4.1 关系运算符和关系表达式	54
4.1.1 关系运算符	54
4.1.2 关系表达式	55
4.2 逻辑运算符和逻辑表达式	56
4.2.1 逻辑运算符	56
4.2.2 逻辑表达式	57
4.3 逻辑 if 语句	57
4.4 块 if 结构	59
4.4.1 单分支选择块 if 结构	59
4.4.2 双分支选择块 if 结构	61
4.4.3 多分支选择块 if 结构	62
4.5 块 if 结构的嵌套	64
4.6 块 case 结构	65
4.7 程序举例	67
习题 4	72
<b>第 5 章 循环结构程序设计</b>	75
5.1 do 循环结构	75
5.1.1 do 循环结构的组成	76
5.1.2 do 循环结构的执行过程	76
5.1.3 do 循环结构嵌套	79
5.1.4 隐含 do 循环结构	80
5.2 do while 循环结构	84
5.2.1 do while 循环结构的组成	84

**教学目标：**

- 了解程序设计与程序设计语言的相关概念。
- 了解 FORTRAN 语言发展简史。
- 了解 Fortran 95 语言的特点。
- 熟悉 Fortran 95 的编译环境。
- 掌握 Fortran 95 编程的上机步骤。

随着计算机科学与技术的发展,人类已步入信息化时代。现在,越来越多的人在利用计算机来处理自己的各类事物,离开计算机可以说是寸步难行。计算机能够完成预定的任务是计算机硬件和软件协同工作的结果,当用户使用计算机完成某项工作时,通常有两种情况:一种情况是借助现成的应用软件来完成,如进行文字处理可使用 Word、WPS Office 等文字处理软件,科学计算可使用 MATLAB、Ansys 等;另一种情况则是没有完全适合现成的应用软件,需要使用某种计算机语言来编制程序完成特定的任务,这就是程序设计。

学习 FORTRAN 程序设计的目的,就是要学会利用 FORTRAN 语言编写出适合实际需要的程序,让计算机完成指定的任务。

本章主要介绍程序设计以及 FORTRAN 语言程序设计的有关知识,使读者对程序设计有一个初步的了解。

## 1.1 程序设计与程序设计语言

程序就是计算机为完成某一个任务所必须执行的一系列指令的集合。

程序是软件的主要表现形式,程序设计是软件实现的主要手段,程序设计语言是程序设计的基本工具。伴随着计算机技术的发展,程序设计语言也经历了一个从低级编程语言到高级编程语言的发展过程。在计算机出现的最初阶段,程序设计是通过机器语言以及后来的汇编语言实现的。汇编语言与机器语言同属于低级语言,其语言结构基本上是面向特定机器指令系统的指令序列,对计算过程的描述是在目标机操作的层次上进行的。因此汇编语言和机器语言严格依赖于特定的指令系统,可移植性差。同时,由于语言的描述层次很低,程序的可读性和可维护性差,代码较长,不适合大型软件的开发。

随着计算机硬件功能和性能的增强,软件的规模和复杂度也日益增加。低级的机器语言和汇编语言显然已不能满足更复杂的软件设计要求。大型程序设计需要更符合人们



描述习惯,更具可读性和可理解性的程序设计语言。同时,随着多种硬件结构的出现,人们也希望程序具有较强的可移植性,可以运行在不同的机器上而不要过分依赖特定的机器指令系统和硬件结构。在这种情况下,高级程序设计语言就应运而生了。

高级程序设计语言的出现标志着形式语言理论和编译理论的突破性进展。高级程序设计语言在与目标机无关的层次对所需计算的问题进行描述,因此它可以屏蔽计算过程的执行细节、突出计算过程的目标和基本过程,便于问题的分析和描述。高级程序设计语言在较高的层次上对计算机的执行过程进行抽象描述,一条高级程序设计语言需要通过编译系统转换成机器指令,其本身与具体的目标机无关,只要编译系统能够生成目标机的指令序列,用高级程序设计语言写成的程序就可以运行在任何计算机上。高级程序设计语言的这些特点,使计算机应用进入了一个新的阶段。大量使用低级语言难以实现的规模大、复杂度高、使用周期长、投入资源多的程序设计任务不断出现并得以完成。这些也反过来促进了高级程序设计语言的发展。

用高级语言编写的程序,称为“源程序”。目前,常用的高级语言有很多种,如 FORTRAN 语言、Pascal 语言、C 语言、Visual Basic、Visual C++ 等,每种高级语言都有解释或编译系统。在本书中我们要学习的 FORTRAN 就是一种需要编译执行的程序设计语言。在输入程序代码后,编译系统将源程序代码编译生成可执行文件,后运行。

## 1.2 FORTRAN 语言发展简史

FORTRAN 语言是世界上被最早正式推广使用的高级程序设计语言,它主要适用于科学和工程问题的数值计算。FORTRAN 是 Formula Translation 的缩写,译为中文是“公式翻译”,意思是指 FORTRAN 是一种易于用与数学公式极其相近的形式来书写数学公式的计算机语言。

1956 年第一个 FORTRAN 语言版本在美国诞生,并在 IBM 704 计算机上运行。随后又相继推出了 FORTRAN II (1958 年)和 FORTRAN IV (1962 年)。1966 年美国标准化协会(American National Standard Institute, ANSI)在 FORTRAN IV 的基础上,制定了两级标准版本:FORTRAN(X3.9-1966)和 FORTRAN(X3.10-1966)。这两个版本分别相当于原来的 FORTRAN IV 和 FORTRAN II,并将 FORTRAN(X3.9-1966)标准名简称为 FORTRAN 66。

1972 年国际标准化组织(International Standard Organization, ISO)接受了美国标准,在稍加修改后公布了 ISO FORTRAN 语言的三级国际标准,即完全级、中间级和基本级。其中完全级相当于 FORTRAN IV,基本级相当于 FORTRAN II,中间级介于 FORTRAN II 和 FORTRAN IV 之间。

FORTRAN IV(即 FORTRAN 66)流行了十几年,几乎统治了所有数值计算领域。但在结构化程序设计方法提出以后,FORTRAN 66 日益不能满足要求,因为 FORTRAN 66 并不是一种结构化的程序设计语言。针对这种情况,1976 年美国标准化协会(ANSI)对 FORTRAN(x3.9-1966)进行了重新修订,其中吸纳了各计算机厂商的建议,新增了不少

功能,并于1978年4月正式公布了新的FORTRAN标准,即FORTRAN(X3.9-1978)。新标准包括一个全集和一个子集,并定名为FORTRAN 77。1980年,FORTRAN 77被ISO接受为国际标准。

FORTRAN 77获得了国际上用户的广泛认可和青睐,大多数计算机系统都配备了FORTRAN 77编译程序。在过去的年代里,FORTRAN 77几乎统治了所有的数值计算领域。

随着计算机技术的飞速发展,继FORTRAN语言之后又出现了一些其他高级语言。尤其是随着面向对象的程序设计方法得到迅速的发展和广泛使用的同时,新推出了一些“面向对象”的程序设计语言。如Visual Basic语言、Visual C++语言、Java语言等。这类新一代的高级语言对FORTRAN语言提出了严峻的挑战。

为了提高FORTRAN语言的使用率和竞争力,1991年8月新一代FORTRAN语言Fortran 90问世。它是一个基于可视化操作平台的FORTRAN标准。

Fortran 90向下兼容FORTRAN 77,并且有了明显的改进和增强,如新增了递归调用、结构体、指针、动态数组、重载函数以及可以实现与其他高级语言混合编程等功能。之后又推出了Fortran 95。

Fortran 95是具有强烈现代特色的语言,总结了现代软件的要求与算法应用的发展,增加了许多现代特征的新概念、新功能、新结构、新形式,FORTRAN语言由此也具有更强的生命力。

### 1.3 Fortran 95语言的特点

Fortran 95最显著的扩充主要有以下7个方面:

- (1) 引入了数组运算,使数组的并行化运算成为可能。
- (2) 提高了数值计算的功能。
- (3) 内在数据类型的参数化。
- (4) 引入用户定义的数据类型,提高了处理能力。
- (5) 引入用户定义的运算和赋值。
- (6) 引入模块数据及过程定义的功能。
- (7) 引入指针概念。

另外,还包括了其他一些扩充。例如,改进了源程序的书写形式、引入了更多的控制结构和递归过程、新的输入输出功能及动态可分配数组等。

Fortran 95的先进性,体现在以下6个方面:

- (1) 增加了许多具有现代特点的项目和语句,用新的控制结构实现选择、分支与循环操作,真正实现了程序的结构化设计。
- (2) 增加了结构块、模块及过程调用的灵活性,使源程序易读易维护。
- (3) 吸收了C、Pascal语言的长处,淘汰或拟淘汰原有过时的语句,加入现代语言的特色。
- (4) 在数值计算的基础上,进一步发挥了计算的优势。新增了许多先进的调用手段,

扩展了操作功能。

(5) 增加了多字节字符集的数据类型及相应的内部函数。允许在字符数据中选取不同种别,在源程序字符串中可以使用各国文字和各种专用符号,对非英语国家使用计算机提供了更大更有效的支持。

(6) FORTRAN 早期版本的程序仍能在 Fortran 95 编译系统下运行,即具有向下兼容性。

## 1.4 程序设计引例

为使读者对 FORTRAN 语言程序和程序设计的基本方法有一个初步的了解,下面通过引例做简单的介绍。

**【例 1-1】** 输入三个数据,计算它们的算术平均值和几何平均值。

分析: 假设用  $a$ 、 $b$ 、 $c$  分别表示三个数,在数学上,要计算这三个数的算术平均值和几何平均值,采用以下公式:

$$\text{算术平均值} = \frac{a+b+c}{3}$$

$$\text{几何平均值} = \sqrt[3]{a \times b \times c}$$

如何编写程序在计算机上实现上述公式的计算呢?

可按照以下步骤进行:

第 1 步: 定义  $a$ 、 $b$ 、 $c$  以及存放计算结果的变量  $ave1$  和  $ave2$ 。

第 2 步: 给  $a$ 、 $b$ 、 $c$  输入数据。

第 3 步: 分别利用公式计算算术平均值和几何平均值。

第 4 步: 输出计算结果。

按照上述方法和步骤,可编写 FORTRAN 语言程序如下:

```
real a,b,c,ave1,ave2           !变量定义说明
read *,a,b,c                   !输入变量 a 和 b 的值
ave1=(a+b+c)/3                 !计算算术平均值
ave2=(a*b*c)**(1.0/3)          !计算几何平均值
print *,"算术平均值为:",ave1  !输出 ave1 的值
print *,"几何平均值为:",ave2  !输出 ave2 的值
end
```

程序运行结果如图 1.1 所示。

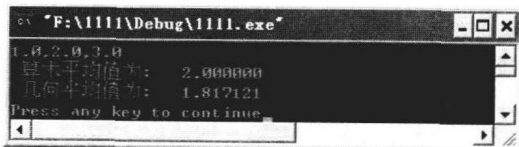


图 1.1 例 1-1 运行结果

程序结构和含义分析：

第 1 行是变量定义(说明)语句,定义了 a、b、c、ave1、ave2 变量为实型变量,目的是在计算机的内存单元中开辟 5 个与变量相对应的存储空间,以便存放要计算的数据和计算结果。

第 2 行是输入语句,执行此语句时,计算机等待用户从键盘输入三个数据分别存放到 a、b、c 变量中。

第 3、4 行是赋值语句,也是计算部分,计算表达式  $(a+b+c)/3.0$ , 求出算术平均值,赋值给变量 ave1,即保存到变量中;计算表达式  $(a * b * c)**(1.0/3)$ , 求出几何平均值,赋值给变量 ave2。程序中的表达式是数学表达式的 FORTRAN 表示方式。

第 5、6 行是输出语句,分别输出保存在变量 ave1 和 ave2 中的算术平均值和几何平均值。

第 7 行是 end 语句,表示程序结束,每一个程序单元结束都要由 end 语句作标志。

另外,程序中“!”后是对程序的注释,注释是非常重要的部分,没有注释不能算合格的程序,通过注释能够使程序更清晰,容易阅读。

通过例 1-1 可以看到,利用程序设计的方法解决问题的基本步骤为:

- (1) 分析问题,建立数学模型(公式)。
- (2) 设计算法,确定功能。
- (3) 选择语言,编写程序。
- (4) 调试程序,输出结果。

其中设计算法是程序设计的主要步骤。

所谓算法,是指为解决给定问题而需要计算机去一步一步执行的有穷操作过程的描述。例 1-1 中给出的实现方法和步骤就是算法。

一个算法必须具有以下特征:

(1) 有穷性。算法的执行步骤总是有限次的,即一个算法必须在执行有穷步后结束,并且,任何算法必须在有限的时间(合理的时间)内完成。显然,一个算法如果永远不能结束或需要运行相当长的时间才能结束,这样的算法是没有使用价值的,如让计算机执行一个历时 100 年才结束的算法,这虽然是有穷的,但超过了合理的限度,也不能把它视作有效算法。

(2) 确定性。算法中的每一步骤必须表达明确的含义,不能有歧义性。例如,两位同学约会,甲对乙说“我在 110 等你”,这个步骤就是不确定的,表现在两个方面,第一是地点上的不确定,到底是在五教 110,还是在八教 110,或者在 110 报警亭;第二就是时间上的不确定,到底是哪一天,几点。这都给对方非常模糊的概念。

(3) 可执行性。算法中的每一个步骤都能被有效地执行,并得到确定的结果。例如:当 B 是一个很小的实数时,  $A/B$  在代数中是正确的,但在算法中是不正确的,它在计算机上无法执行,要使  $A/B$  能正确执行,必须在算法中控制 B 满足条件:  $|B| > \delta$ ,  $\delta$  是一个计算机允许的合理小实数。

(4) 输入的数据大于等于 0。一般的程序都会要求若干个输入信息,即要加工处理的“原料”。但是,有些特殊提名的“原料”也可以在程序中自动产生,此时可以没有输入。

(5) 至少有一个数据输出。算法的目的是为了解决一个给定的问题,解决的最终目的就是给出最后的结果,即“解”,所谓“解”就是输出。算法在执行过程中必须有输出的操作,即算法中必须有输出数据的步骤。但是算法的输出不一定是计算机的打印输出,一个算法得到的结果就是算法的输出。没有输出步骤的算法是毫无意义的。

算法的表示方法很多,常用的有自然语言和流程图。

自然语言就是人们日常使用的语言,可以是汉语、英语或其他语言,从根本上讲,程序就是用计算机语言描述的算法。

算法的另一种表示方法是采用图框的方法描述算法。即流程图法。它用不同的几何图形来代表不同性质的操作,例如,用矩形框表示要进行的操作,用菱形框表示判断,用流程线将各步操作连接起来并指向算法的执行方向。其优点是描述简洁、清晰和直观,缺点是由于转移箭头的无约束使用,影响算法的可靠性。图 1.2 和图 1.3 是用流程图描述程序的三种基本结构。

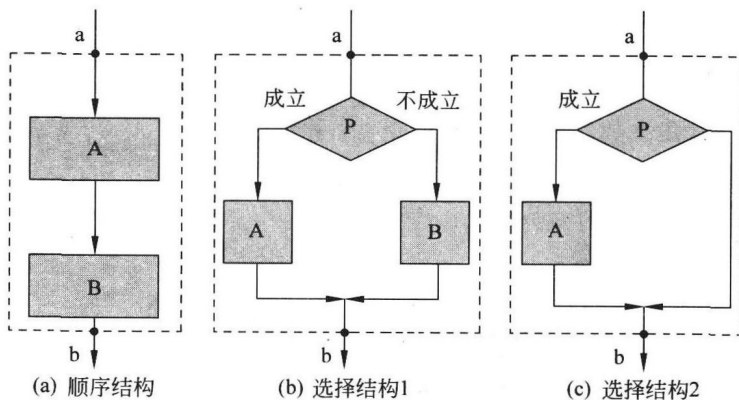


图 1.2 顺序结构和选择结构

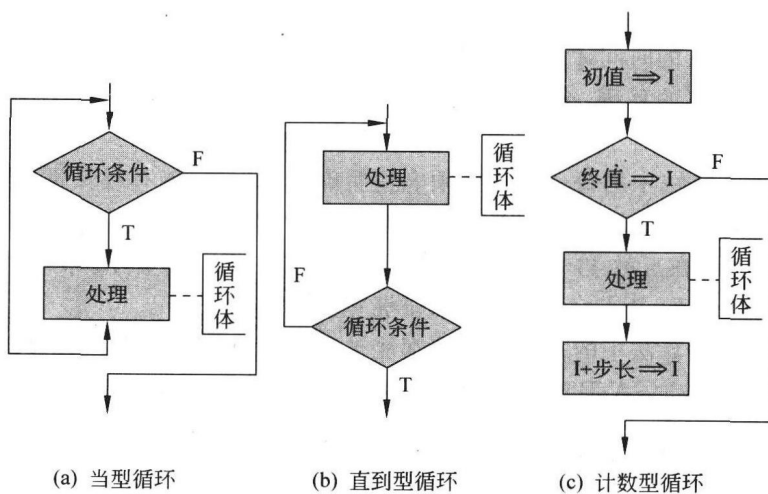


图 1.3 循环结构

算法是程序设计的核心和基础。算法构造的思维方法与一般数学系统的思维方法有所不同,理解、熟悉和习惯算法构造的思维方法,是学习计算机程序设计的基本内容。从某种意义上说,培养算法设计能力实际上就是培养合理进行计算的能力,而要发现这种合理性,寻得“简洁算法”,首先就必须有很好的观察能力和对基础知识的良好掌握。

另一方面,程序的处理对象是数据,每个数据都有一定的特性和关联,因此为了更好地处理和操作,要研究数据,对数据进行有规律的组织 and 构造。在算法设计中,数据的组织和构造都有其基本方式和规律,这种组织和构造方式称为数据结构。对于不同的数据结构,程序中要采用不同的处理方法。所以,就像前面已经提到的,程序主要描述两部分内容,算法和数据结构。著名的计算机科学家 Wirth 提出了一个著名的公式来表达程序的实质:

$$\text{程序} = \text{数据结构} + \text{算法}$$

算法设计好后,选择一种程序设计语言来描述算法,即编写程序,得到程序代码,例如例 1-1 的源程序。要正确合理地编写程序,必须掌握程序语言的语句、函数、结构、语法等。在本书中,我们要在后续章节中陆续介绍 FORTRAN 语言的数据结构、语句、函数、结构、语法等以及一些相关的常用算法。

## 1.5 Fortran 95 编译环境与上机步骤

要编写并运行程序,需要相应的开发工具。最早微软公司推出 Microsoft Fortran PowerStation 4.0 开发环境用于 Visual Fortran 的开发。在 1997 年 3 月微软和数据设备公司(Digital Equipment Corp, DEC)合作研究,开发和推出了 Digital Visual Fortran 5.0,1998 年 1 月,DEC 与 Compaq 公司合并,DEC 成为 Compaq 公司的全资子公司,其后又推出 Compaq Visual Fortran 等。

下面以例 1-1 为例,以 Compaq Visual Fortran 6.5 为集成开发环境,来介绍上机的基本知识与步骤,包括进入和退出环境,了解常用界面,建立工作空间、项目(工程)、文件、存储与运行程序等。

### 1.5.1 Compaq Visual Fortran 6.5 的安装与启动

Compaq Visual Fortran 6.5 的安装方法与其他应用程序安装类似。

最常见的启动方法是:选择“开始”|“所有程序”|Compaq Visual Fortran 6.5 | Developer Studio,如图 1.4 所示,就可启动 Compaq Visual Fortran 6.5,启动后将看到如图 1.5 所示的工作窗口。

### 1.5.2 上机步骤

FORTRAN 程序上机运行一般要经过编写(辑)程序、编译、连接和运行 4 个步骤。

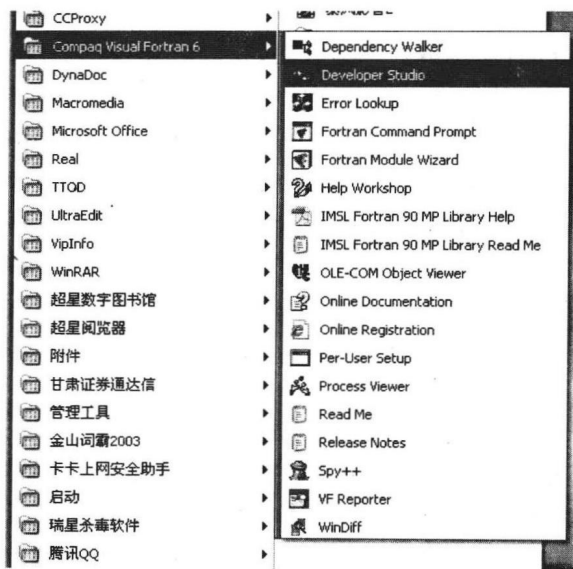


图 1.4 Compaq Visual Fortran 6.5 启动过程

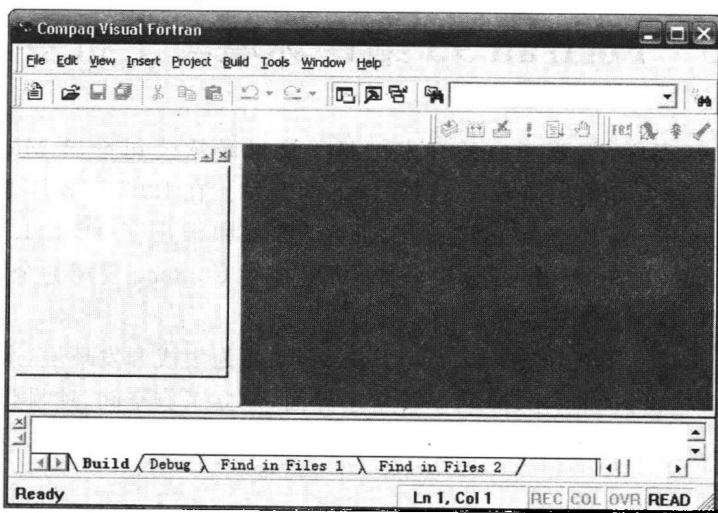


图 1.5 Compaq Visual Fortran 6.5 工作窗口

具体操作如下：

### 1. 创建工作空间

在第一次创建 FORTRAN 程序时首先应在磁盘上建立一个工作空间，即创建一个文件夹，文件夹装有两个管理文件，通过工作空间来合理地组织管理项目和文件。这里建立一个名为 forpro 的工作空间。

创建步骤如下：

- ① 选择 File|New 命令，如图 1.6 所示，弹出 New 对话框，选择 Workspaces 选项卡。

② 在名称和位置文本框中分别输入工作空间名和路径。路径可以通过单击右侧的按钮打开浏览窗口查找和定位。

③ 单击 OK 按钮创建完成新的工作空间,回到 Developer Studio 主窗口,如图 1.7 所示。

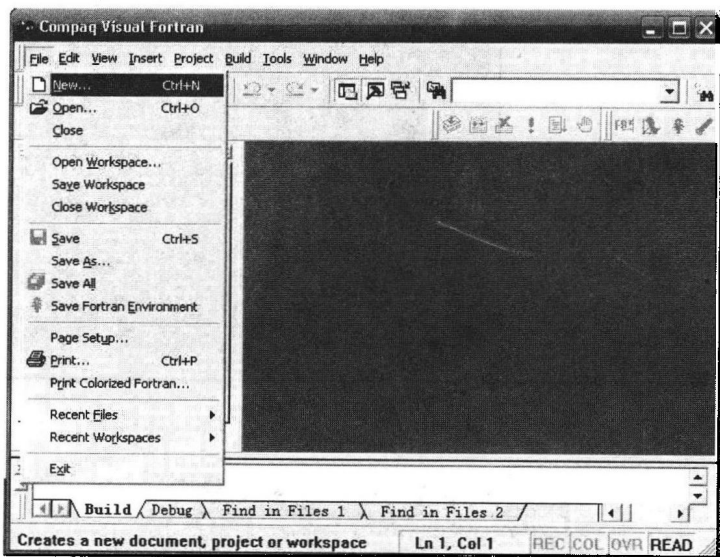
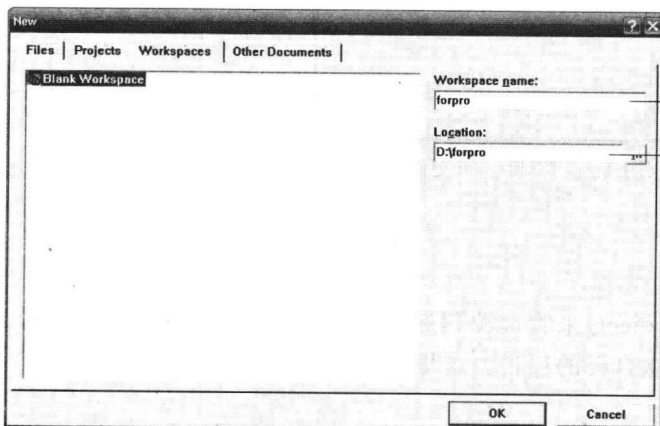


图 1.6 创建工作窗口



在对话框中选择Workspaces选项卡  
在location中选择工作空间要建立的位置,在Workspace name文本框中输入工作空间名称。然后单击OK按钮

图 1.7 创建工作窗口

完成建立新的工作空间后,会在 Developer Studio 主窗口的工作空间管理窗口内建立新的选项卡 File View,同时显示“Workspace ‘forpro’ : 0 Project(s)”,指出工作空间名称和项目数,如图 1.8 所示。在 D 盘上创建了新的文件夹 D: \forpro,并且生成两个工作空间管理文件 forpro. opt 和 forpro. dsw。以后要打开工作空间 forpro 也可以直接双击 forpro. dsw 文件,如图 1.9 所示。



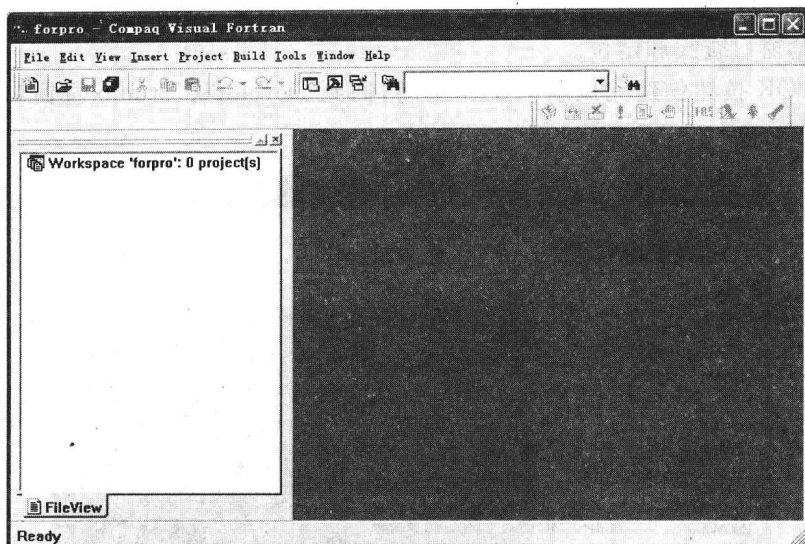


图 1.8 创建工作空间



图 1.9 项目空间文件夹

## 2. 创建项目空间

开发工具通过项目(工程, Project)来管理源程序文件, 并一起作为编译程序单位, 因此建立工作空间后, 要在其中建立自己的项目。这里建立项目 exam1。

创建步骤如下:

① 再次打开 New 对话框, 选取 Projects 选项卡, 选择应用程序类型为 Fortran Console Application, 如图 1.10 所示。

② 选中 Add to current workspace 单选按钮, 在名称文本框中输入项目名称, 如图 1.11 所示。

③ 单击 OK 按钮创建完成新的项目, 回到 Developer Studio 主窗口。

Visual Fortran 6.5 以前的版本不会出现如图 1.12 所示的界面, 旧版本直接跳过。建议选用第一个选项 An empty project, 然后直接单击 Finish 按钮。