



国家级实验教学示范中心联席会计算机学科规划教材  
教育部高等学校计算机类专业教学指导委员会推荐教材  
面向“工程教育认证”计算机系列课程规划教材

# FORTRAN语言程序设计

## ——FORTRAN95

◎ 王丽娟 段志东 主编  
李玉龙 主审



清华大学出版社

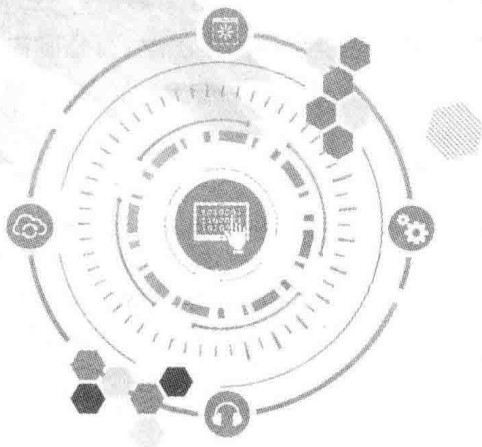


国家级实验教学示范中心联席会计算机学科规划教材  
教育部高等学校计算机类专业教学指导委员会推荐教材  
面向“工程教育认证”

# FORTRAN语言程序设计

## ——FORTRAN95

◎ 王丽娟 段志东 主编



清华大学出版社  
北京

## 内 容 简 介

本书全面、系统地介绍 FORTRAN95 的语法规则以及利用它进行程序设计的方法。主要内容有：FORTRAN95 概述及编译环境的介绍、FORTRAN95 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。另外根据教学需要，本书扩充了计算思维和计算机系统组成的相关知识。

本书针对程序设计初学者的特点，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。本书可作为高等院校理工科学生学习程序设计的教材，也可以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

FORTRAN 语言程序设计：FORTRAN95/王丽娟，段志东主编. —北京：清华大学出版社，2017  
(面向“工程教育认证”计算机系列课程规划教材)

ISBN 978-7-302-48390-8

I. ①F… II. ①王… ②段… III. ①FORTRAN 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 217459 号

责任编辑：付弘宇

封面设计：刘 健

责任校对：焦丽丽

责任印制：李红英

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969，[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

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

课 件 下 载：<http://www.tup.com.cn>，010-62795954

印 装 者：三河市铭诚印务有限公司

经 销：全国新华书店

开 本：185mm×260mm

印 张：22

字 数：533 千字

版 次：2017 年 9 月第 1 版

印 次：2017 年 9 月第 1 次印刷

印 数：1~2000

定 价：49.00 元



# 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,使其掌握高级语言程序设计的基本思想和方法,培养学生的计算思维,增强其用计算机处理问题的能力。

FORTRAN 语言是最早出现的计算机高级程序设计语言,其发展过程中不断吸收现代化编程语言的新特性,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使用 FORTRAN 语言编写的,如 ANSYS、Marc 等。

基于 Windows 平台下的 FORTRAN90 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(命令行界面),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 FORTRAN95 为平台,重点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知识点,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。

为使读者更好地掌握 FORTRAN95 程序设计基础,我们还编写了与本书配套的《FORTRAN95 程序设计实验指导及测试》,可作为学习参考书。另外,还有与本书配套的电子版教学课件,供教师教学参考使用。

书中所有程序实例都是由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,使读者在程序设计的思维训练和程序组织方面得到极大简化。为适应不断更新的计算机操作系统,在实验教材中还给出了 Windows 7、Windows 10 操作系统下使用 Visual Fortran 的上机操作过程。

本书可作为高等学校理工科学生学习程序设计的教材,也可以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。

本书由王丽娟、段志东主编,李玉龙主审。第 1、2、12、14 章由王红鹰编写,第 3、5、6、7 章由陈权编写,第 9、11、13 章和附录 A 由段志东编写,第 4、8、10 章和附录 B 由王丽娟编写。与本书配套的《FORTRAN95 程序设计实验指导及测试》一书由王红鹰、陈权主编,李玉龙主审。

本书在规划、编写过程中得到了兰州交通大学教务处、计算机教学示范中心、电信学院、



继续教育学院、土木工程学院的领导和教师们的大力支持,编者在此表示衷心感谢。

鉴于编者水平所限,书中难免有不当或错误之处,恳请读者不吝赐教。

本书的配套电子课件等资源可以从清华大学出版社网站 [www.tup.com.cn](http://www.tup.com.cn) 下载,在本书和课件的使用中如有问题,请联系 [fuhy@tup.tsinghua.com.cn](mailto:fuhy@tup.tsinghua.com.cn)。

编 者

2017年6月



# 目 录

<b>第 1 章 计算思维与程序设计</b> .....	1
1.1 什么是计算 .....	1
1.1.1 计算机的硬件 .....	1
1.1.2 计算 .....	2
1.1.3 计算机的软件 .....	3
1.1.4 计算科学 .....	3
1.2 什么是计算思维 .....	4
1.2.1 计算思维的基本原则 .....	4
1.2.2 计算思维的基本应用 .....	5
1.2.3 计算思维在日常生活中的体现 .....	6
1.3 计算机语言 .....	7
1.3.1 机器语言 .....	7
1.3.2 汇编语言 .....	8
1.3.3 高级语言 .....	8
1.4 算法 .....	10
1.4.1 算法的特征 .....	10
1.4.2 算法评价指标 .....	11
1.4.3 算法的表示 .....	11
1.5 程序设计 .....	15
1.5.1 程序设计步骤 .....	15
1.5.2 程序设计方法 .....	16
习题 1 .....	17
<b>第 2 章 FORTRAN95 概述</b> .....	18
2.1 FORTRAN 语言发展概况 .....	18
2.2 FORTRAN95 语言的特点 .....	19
2.3 简单的 FORTRAN95 程序分析 .....	20
2.4 FORTRAN95 编译环境与上机步骤 .....	23



2.4.1	Compaq Visual Fortran 6.5 的安装与启动 .....	23
2.4.2	上机步骤 .....	24
习题 2	.....	34
<b>第 3 章</b>	<b>FORTRAN95 程序设计初步</b> .....	<b>35</b>
3.1	FORTRAN95 的字符集、标识符和关键字 .....	35
3.1.1	字符集 .....	35
3.1.2	标识符 .....	35
3.1.3	关键字 .....	36
3.2	FORTRAN95 程序的书写格式 .....	36
3.2.1	固定格式 .....	36
3.2.2	自由格式 .....	37
3.3	FORTRAN95 的数据类型 .....	37
3.3.1	数值型数据的表示及存储 .....	37
3.3.2	非数值型数据的表示及存储 .....	38
3.4	常量和变量 .....	39
3.4.1	常量 .....	39
3.4.2	变量 .....	44
3.5	FORTRAN95 的算术运算符与算术表达式 .....	48
3.5.1	算术运算符 .....	48
3.5.2	算术表达式 .....	49
3.6	FORTRAN95 标准函数 .....	51
习题 3	.....	53
<b>第 4 章</b>	<b>顺序结构程序设计</b> .....	<b>55</b>
4.1	赋值语句 .....	56
4.1.1	赋值语句的性质和作用 .....	56
4.1.2	执行算术赋值语句时的类型转换问题 .....	57
4.1.3	字符赋值语句和字符运算符 .....	58
4.2	简单的输出语句 .....	59
4.2.1	输出语句的作用和分类 .....	59
4.2.2	表控输出语句 .....	59
4.3	简单的输入语句 .....	60
4.3.1	输入语句的作用和分类 .....	60
4.3.2	表控输入语句 .....	61
4.4	END 语句、STOP 语句和 PAUSE 语句 .....	63
4.4.1	END 语句 .....	63
4.4.2	STOP 语句 .....	63
4.4.3	PAUSE 语句 .....	63



4.5 程序举例 .....	64
习题 4 .....	66
<b>第 5 章 选择结构程序设计 .....</b>	<b>68</b>
5.1 选择结构中的条件准备 .....	68
5.1.1 关系运算符和关系表达式 .....	68
5.1.2 逻辑运算符和逻辑表达式 .....	70
5.2 逻辑 IF 语句 .....	72
5.3 块 IF 结构 .....	73
5.3.1 单分支块 IF 结构 .....	73
5.3.2 双分支选择块 IF 结构 .....	74
5.3.3 多分支块 IF 结构 .....	75
5.4 块 IF 结构的嵌套 .....	77
5.5 块 CASE 结构 .....	78
5.6 程序举例 .....	80
习题 5 .....	85
<b>第 6 章 循环结构程序设计 .....</b>	<b>88</b>
6.1 用 DO 语句实现循环结构 .....	88
6.1.1 循环语句(DO 语句)和循环次数的计算 .....	89
6.1.2 DO 循环的执行过程 .....	92
6.1.3 DO 循环的一些说明 .....	96
6.1.4 DO 循环结构的嵌套 .....	98
6.1.5 隐含 DO 循环结构 .....	103
6.2 DO WHILE 循环结构 .....	106
6.2.1 DO WHILE 循环的组成 .....	106
6.2.2 DO WHILE 循环的执行过程 .....	107
6.3 循环的流程控制 .....	109
6.3.1 EXIT 语句 .....	109
6.3.2 CYCLE 语句 .....	110
6.4 几种循环形式的关系和比较 .....	110
6.5 程序举例 .....	112
习题 6 .....	115
<b>第 7 章 格式输入和输出 .....</b>	<b>117</b>
7.1 格式输出 .....	117
7.1.1 I 编辑符 .....	117
7.1.2 F 编辑符 .....	119
7.1.3 E 编辑符 .....	120





7.1.4	D 编辑符	122
7.1.5	L 编辑符	122
7.1.6	A 编辑符	123
7.1.7	撇号编辑符	123
7.1.8	X 编辑符	124
7.1.9	斜杠编辑符	125
7.1.10	重复系数	125
7.1.11	WRITE(PRINT)语句与 FORMAT 语句的相互作用	125
7.2	格式输入	128
7.2.1	格式输入的一般形式	128
7.2.2	整数的输入	128
7.2.3	实数、复数和双精度数的输入	129
7.2.4	逻辑型数据的输入	130
7.2.5	字符型数据的输入	131
7.2.6	对格式输入的说明	132
7.3	在输入输出语句中包含格式说明	133
	习题 7	134
<b>第 8 章</b>	<b>数组</b>	<b>136</b>
8.1	数组的概念	138
8.2	数组的说明	139
8.2.1	用 DIMENSION 语句定义数组	139
8.2.2	用类型说明语句定义数组	140
8.2.3	同时使用类型说明符和 DIMENSION 语句定义数组	140
8.3	给数组赋初值	140
8.3.1	使用数组赋值符赋初值	140
8.3.2	用 DATA 语句给数组赋初值	141
8.4	对数组的操作	142
8.4.1	对数组元素的操作	142
8.4.2	数组的整体操作	143
8.4.3	数组局部引用	143
8.4.4	WHERE 命令	144
8.4.5	FORALL 命令	145
8.5	数组的存储规则	146
8.5.1	一维数组的存储规则	146
8.5.2	二维数组的存储规则	147
8.5.3	三维数组的存储规则	147
8.6	数组的输入和输出	148
8.6.1	用 DO 循环结构输入输出数组	148

8.6.2	用数组名作为输入输出项 .....	150
8.6.3	用隐含 DO 循环输入输出数组 .....	151
8.7	动态数组 .....	153
8.8	数组应用举例 .....	154
8.8.1	一维数组程序举例 .....	155
8.8.2	二维数组程序举例 .....	163
习题 8	.....	167
<b>第 9 章</b>	<b>函数与子程序 .....</b>	<b>169</b>
9.1	概述 .....	169
9.1.1	子程序产生的原因 .....	169
9.1.2	FORTRAN 子程序 .....	176
9.2	语句函数 .....	178
9.2.1	语句函数的定义 .....	180
9.2.2	语句函数的调用 .....	182
9.2.3	语句函数应用举例 .....	183
9.3	函数子程序 .....	185
9.3.1	函数子程序的定义 .....	186
9.3.2	函数子程序的调用 .....	187
9.4	子例行子程序 .....	189
9.4.1	子例行子程序的定义 .....	190
9.4.2	子例行子程序的调用 .....	190
9.5	程序单元之间的数据传递:虚实结合 .....	192
9.5.1	简单变量作为虚参时的虚实结合 .....	193
9.5.2	数组作为虚参时的虚实结合 .....	195
9.5.3	子程序名作为虚参时的虚实结合 .....	199
9.5.4	星号(*)作为虚参 .....	200
9.5.5	子程序中变量的生存周期 .....	200
9.6	特殊的子程序类型 .....	201
9.6.1	递归子程序 .....	201
9.6.2	内部子程序 .....	207
9.7	数据共用存储单元与数据块子程序 .....	208
9.7.1	等价语句 .....	208
9.7.2	公用语句 .....	211
9.7.3	数据块子程序 .....	215
习题 9	.....	217
<b>第 10 章</b>	<b>文件 .....</b>	<b>220</b>
10.1	文件的基本概念 .....	220



10.1.1	记录	220
10.1.2	文件的概念	221
10.1.3	文件的特性	221
10.1.4	文件的定位	222
10.2	文件的操作语句	223
10.2.1	文件的打开与关闭	223
10.2.2	文件的输入(read)语句和输出(wirte)语句	226
10.2.3	查询文件的状态(inquire)语句	227
10.2.4	rewind 语句	228
10.2.5	backspace 语句	228
10.2.6	endfile 语句	229
10.3	有格式文件的存取	229
10.3.1	有格式顺序文件存取	229
10.3.2	有格式直接文件存取	230
10.4	无格式文件的存取	232
10.4.1	无格式顺序文件存取	232
10.4.2	无格式直接文件存取	233
10.5	二进制文件的存取	234
10.5.1	二进制顺序文件存取	234
10.5.2	二进制直接文件存取	235
习题 10		236
<b>第 11 章</b>	<b>派生数据类型与结构体</b>	<b>238</b>
11.1	概述	238
11.2	派生类型定义	240
11.3	结构体的定义与引用	241
11.3.1	结构体定义	241
11.3.2	结构体成员引用	242
11.4	结构体初始化	242
11.4.1	用赋值语句给结构体成员赋值	242
11.4.2	定义的同时给结构体成员赋值	243
11.5	结构体数组	244
11.5.1	结构体数组定义	244
11.5.2	结构体数组初始化	245
11.6	程序举例	245
习题 11		254
<b>第 12 章</b>	<b>指针</b>	<b>256</b>
12.1	指针的概念	256

12.2	指针的定义与使用 .....	257
12.3	指针的使用 .....	258
12.3.1	指向一般变量的应用 .....	258
12.3.2	指向动态存储空间 .....	260
12.4	指针与数组 .....	262
12.4.1	指针指向其他数组 .....	262
12.4.2	指针指向动态配置的内存空间 .....	264
12.5	指针与链表 .....	266
12.5.1	结点的定义 .....	266
12.5.2	链表的基本操作 .....	267
12.5.3	综合实例 .....	273
习题 12	.....	277
<b>第 13 章</b>	<b>模块、接口与重载 .....</b>	<b>279</b>
13.1	模块的定义 .....	280
13.2	USE 语句 .....	281
13.3	接口界面块 .....	285
13.4	重载 .....	287
13.4.1	函数和子例程序的重载 .....	287
13.4.2	赋值号重载 .....	291
13.4.3	操作符重载 .....	292
13.5	应用举例 .....	293
习题 13	.....	299
<b>第 14 章</b>	<b>常用数值算法 .....</b>	<b>300</b>
14.1	求解一元方程 .....	300
14.1.1	二分法 .....	300
14.1.2	弦截法 .....	302
14.1.3	迭代法 .....	303
14.1.4	牛顿迭代法 .....	305
14.2	数值积分 .....	307
14.2.1	矩形法 .....	307
14.2.2	梯形法 .....	308
14.2.3	辛普生法 .....	311
14.3	线性代数 .....	314
14.3.1	矩阵的加、减、乘法运算 .....	314
14.3.2	三角矩阵 .....	316
14.3.3	Gauss-Jordan 法求联立方程组 .....	318



14.4 求解常微分方程 .....	320
习题 14 .....	322
附录 A ASCII 码字符编码表 .....	324
附录 B FORTRAN 库函数 .....	328
参考文献 .....	336

教学目标:

- 掌握计算机的系统组成;
- 理解计算思维的概念及应用;
- 理解三种计算机语言及其相互间的关系;
- 掌握算法的概念及其表示方式;
- 掌握三种基本结构;
- 理解程序和程序设计的含义。

计算机科学是关于计算的学科,计算是利用计算机解决问题的过程。计算思维就是计算机科学家在用计算机解决问题时形成的特有的思维方式和解决方法。

### 1.1 什么是计算

#### 1.1.1 计算机的硬件

计算机是 20 世纪人类最伟大的发明之一。自从世界上第一台电子数字计算机诞生以来,在短短的几十年内得到了迅速发展和广泛应用。现在计算机已经应用到社会、生活的几乎每一个方面。人们利用计算机上网冲浪、写文章、打游戏或听歌、看电影,机构用计算机管理企业、设计制造产品或从事电子商务,大量仪器设备由计算机控制,手机与电脑之间的差别越来越小,……总之计算机似乎是无处不在、无所不能的。那么,计算机是如何做到这一切的呢?为了回答这个问题,需要了解计算机系统构成和计算机的工作原理。

提到计算机,人们的脑中首先会浮现出显示器、键盘、鼠标、主机箱等一堆计算机硬件设备。下面先来了解一些计算机硬件设备的基础知识,学习用计算机解决问题的计算机制。现代计算机的主要功能部件如图 1.1 所示。

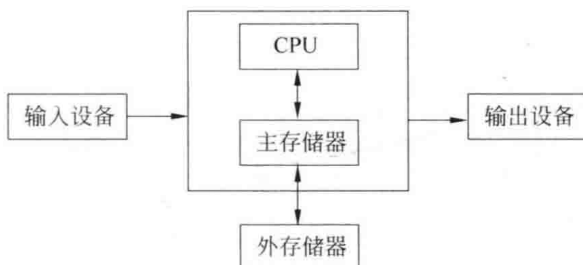


图 1.1 计算机的基本结构(主要功能部件)

## 1. CPU、指令与程序

中央处理器(CPU)是计算机的核心部件,主要包括运算器和控制器。CPU 是计算机的计算部件,能够执行机器指令(简称指令, instruction)。指令由操作码和操作数构成,每条指令表达的是计算机对特定数据执行特定操作。某种 CPU 能执行的全体指令是由该 CPU 制造商设计并保持固定不变的,称为该 CPU 的指令集。例如, Intel 公司为它的 80x86 系列处理器设计了上百条的指令。

外行人也许认为,计算机的功能如此强大是因为它能执行功能强大的指令,然而事实并非如此。即使是目前最先进、计算能力最强大的计算机,它的 CPU 也只会执行一些非常简单的指令,例如两个数相加、判断两个数是否相等、把数据放入指定的存储单元,等等。

由于每条指令都只能完成很简单的操作,因此只靠少数几条指令是无法完成复杂的事情的。但是,如果将成千上万条简单指令组合起来,就能解决非常复杂的问题。所以说,复杂操作可以通过执行按特定次序排列的许多简单操作而实现。这种由许多指令按次序排列而成并交给计算机逐条执行的指令序列就称为程序(program)。为了用计算机解决问题,把问题的解法表达成一个指令序列(即程序)的过程,称为程序设计或编程(programming)。可见,计算机所做的一切神奇的事情,都是靠一步一步执行平凡而乏味的简单指令序列来做到的。计算机一点也不神奇,它唯一会做的事情就是机械地执行预定的指令序列。

## 2. 存储器

存储器是计算机的记忆部件,用于存储数据和程序。

存储器分为主存储器和外存储器,它们是用不同的物理材料制造的。CPU 只能直接访问主存储器,也只有主存储器才能提供与 CPU 相匹配的存取速度。主存储器需要持续供电来维持存储,一旦断电,存储的数据或程序就会消失。为了长期、持久地存储信息,可以使用即使断电也能保持存储的外存储器,如硬盘。CPU 不能直接访问外存储器,外存储器上的数据或程序必须先导入到主存储器中,才能被 CPU 存取或执行。外存储器的读写速度远远低于主存储器,这个差别极大地影响了计算机解决问题时所使用的方法。

## 3. 输入输出设备

输入和输出设备是人与计算机进行交互的设备。我们通过输入设备向计算机输入信息,计算机则通过输出设备将计算结果告知我们。传统的输入设备有键盘和鼠标等,输出设备有显示器和打印机等。现代的触摸屏则兼具输入和输出功能。

现代计算机在体系结构上的特点是:数据和程序都以二进制形式存储在主存储器中,CPU 通过访问主存储器来取得待执行的指令和待处理的数据。这称为冯·诺依曼(John von Neumann, 1903—1957)体系结构。

### 1.1.2 计算

了解了计算机的组成,就能理解计算机解决问题的过程。我们来看一个计算机常见任务——写文章是如何实现的。为了解决这个问题,首先需要编写具有输入、编辑、保存文章等功能的程序,如微软公司的 Word 程序、金山公司的 WPS 文字程序等。如果这个 Word 程序已经存入(通过安装)计算机的外存储器(硬盘),通过双击 Word 程序图标等方式可以启动这个程序,也就是将该程序从硬盘加载到内存储器中。然后 CPU 逐条取出该程序的指令并执行,直至最后一条指令执行完毕,程序即告结束。在程序执行过程中,有些指令会



导致与用户的交互,例如,用户利用键盘输入或删除文字,利用鼠标选择菜单进行存盘或打印,等等。就这样,通过执行成千上万条简单的指令,最终解决了计算机写文章的问题。

针对一个问题,设计出解决问题的程序(指令序列),并由计算机来执行这个程序,这就是计算(computation)。

通过计算,使得只会执行简单操作的计算机能够完成复杂任务,所以计算机的神奇表现其实都是计算的威力。我们再通过一个简单的例子来了解计算的能力。小莉是一个只学过加法的一年级小学生,她能完成一个乘法运算的任务吗?答案是肯定的。解决问题的关键在于编写出适合的指令序列让小莉机械地执行。例如,下列“程序”就能使小莉计算出  $m \times n$ :

在纸上写下 0,记住结果;

给所记结果加上第 1 个  $n$ ,记住结果;

给所记结果加上第 2 个  $n$ ,记住结果;

.....

给所记结果加上第  $m$  个  $n$ ,记住结果。到这里就得到了  $m \times n$ 。

可以看出,这个指令序列的每一步都是小莉能够做到的,因此最后确实能完成乘法计算。这就是“计算”所带来的成果。

计算机就是通过这样的“计算”来解决所有复杂问题的。执行大量简单指令组成的程序虽然枯燥烦琐,但计算机作为一种机器,其特长正在于机械地、忠实地、不厌其烦地执行大量简单指令。

### 1.1.3 计算机的软件

我们已经知道计算机就是进行“计算”的机器。这里的“计算”显然已不是日常所说的数学计算。事实上,计算机在屏幕上显示信息,在 Word 文档中查找并替换文本,播放 MP3 音乐,这些都是计算。

理解了计算机是如何计算的,也就能理解为什么计算机能解决各种不同类型的问题。其中的奥秘就是程序。如果想用计算机写文章,就将 Word 之类的程序加载到主存储器中让 CPU 去执行;如果想用计算机听音乐,就将 Media Player 之类的程序加载到主存储器中让 CPU 去执行;如果将 Internet Explorer 之类的程序加载到主存储器中让 CPU 去执行,计算机就可以在互联网上浏览信息。一台计算机的硬件虽然固定不变,但通过加载执行不同的程序,就能实现不同的功能,解决不同的问题。

我们平时所说的计算机就是安装了各种不同程序的计算机,这才是一个完整的计算机系统。所以说计算机系统包括了硬件系统和软件系统两部分,其中软件系统就是为运行、管理和维护计算机而编制的各种程序、数据和文档的总称。软件系统又包括系统软件和应用软件,软件的主要构成就是程序。这样的计算机具有通用性。

在工业控制和嵌入式设备等领域存在专用计算机,它们只执行预定的程序,从而实现固定的功能。例如,智能电饭锅其实就是能执行预定程序的计算机。

### 1.1.4 计算科学

为了更好地利用计算机解决问题,人们深入研究了关于计算的理论、方法和技术,形成

了专门研究计算的学科——计算机科学(Computer Science)。

计算机科学包含很多内容,是一门包含各种各样与计算和信息处理相关主题的系统学科,从抽象的算法分析、形式化语法等,到更具体的编程语言、程序设计、软件和硬件等。计算机科学包括理论计算机科学和应用计算机科学。本书侧重于计算机科学家在用计算机解决问题时建立的一些基本思想和方法,这些思想和方法普遍存在于计算机科学的各个分支中。计算机科学家在思考一个根本问题:到底什么是计算机可计算的?一般人会认为,一个问题能不能用计算机计算,取决于该计算机的计算能力;而计算机的计算能力又取决于 CPU 的运算速度、指令集、主存储器容量等硬件指标。然而,作为计算机科学理论基础的可计算理论却揭示出了一个出人意料的结论:所有计算机的计算能力都是一样的!尽管不同计算机有不同的指令集和不同性能的硬件,在计算的时间和空间效率上可能有所差异,但现有的各种计算设备在计算的能力上是等同的。一台计算机能解决的问题,另一台计算机肯定也能解决。

作为一个学科,计算机科学涵盖了从算法的理论研究和计算的极限,到如何通过硬件和软件实现计算系统。计算机科学评审委员会(Computing Sciences Accreditation Board, CSAB)由 Association for Computing Machinery(ACM)和 IEEE Computer Society(IEEE-CS)的代表组成,确立了计算机科学学科的 4 个主要领域:计算理论、算法与数据结构、编程方法与编程语言,以及计算机元素与架构。CSAB 还确立了其他一些重要领域,如软件工程、人工智能、计算机网络与通信、数据库系统、并行计算、分布式计算、人机交互、机器翻译、计算机图形学、操作系统,以及数值和符号计算。

## 1.2 什么是计算思维

如前所述,计算是利用计算机一步一步地执行指令来解决问题的过程,计算机科学是关于计算的科学。正如数学家在证明数学定理时有独特的数学思维、工程师在进行设计制造生产时有独特的工程思维、艺术家在创作诗歌音乐绘画时有独特的艺术思维一样,计算机科学家在用计算机解决问题时也有自己独特的思维方式和解决方法,统称为计算思维。

计算思维的提出,最早可回溯到美国麻省理工学院(MIT)的西蒙·帕佩特(Seymour Papert)教授。美国卡内基·梅隆大学的计算机科学系主任周以真教授则对其进行了系统阐述和推广。2006 年 3 月,周以真(Jeannette M. Wing)教授在美国计算机权威期刊 *Communications of the ACM* 上提出并定义了计算思维(Computational Thinking)。周教授认为,“计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。”

从问题的计算机表示、算法设计直到编程实现,计算思维贯穿于计算的全过程。学习计算思维,就是学会像计算机科学家一样思考和解决问题。

### 1.2.1 计算思维的基本原则

用计算机解决问题时必须遵循的基本原则是:既要充分利用计算机的计算和存储能力,又不能超出计算机的能力范围。计算思维是建立在计算机的能力和限制之上的,这是计算思维区别于其他思维方式的一个重要特征。