

通向金牌之路

# 全国青少年信息学竞赛 培训教材

## Pascal语言程序设计

陈颖 邱桂香 编著

# 全国青少年信息学竞赛培训教材

## ——Pascal 语言程序设计

陈 颖 邱桂香 编著



ZHEJIANG UNIVERSITY PRESS  
浙江大学出版社

## 内容简介

本书结合中学信息学奥赛,贯穿问题导引、分析和程序设计过程,让学习者学会分析问题、设计算法,并使用 Pascal 语言工具描述和实现算法。本书附带光盘提供例题测试数据和练习中非编程题答案。

### 图书在版编目(CIP)数据

全国青少年信息学竞赛培训教材. Pascal 语言程序设计 / 陈颖,邱桂香编著. —杭州:浙江大学出版社, 2010.6  
ISBN 978-7-308-07633-3

I. ①全… II. ①陈… ②邱… III. 计算机课—中小学—教学参考资料 IV. ①G634.673

中国版本图书馆 CIP 数据核字 (2010)第 099712 号

## 全国青少年信息学竞赛培训教材 ——Pascal 语言程序设计

陈颖 邱桂香 编著

---

责任编辑 黄娟琴  
封面设计 刘依群  
出版发行 浙江大学出版社  
(杭州市天目山路 148 号 邮政编码 310007)  
(网址: <http://www.zjupress.com>)  
排 版 杭州中大图文设计有限公司  
印 刷 杭州浙大同力教育彩印有限公司  
开 本 787mm×1092mm 1/16  
印 张 23  
印 数 0001—8000  
字 数 545 千  
版 次 2010 年 6 月第 1 版 2010 年 6 月第 1 次印刷  
书 号 ISBN 978-7-308-07633-3  
定 价 45.00 元(含光盘)

---

版权所有 翻印必究 印装差错 负责调换

浙江大学出版社发行部邮购电话 (0571)88925591

## 丛书编委会

(按姓氏拼音排列)

- 陈颖** 福州第一中学教师,国际信息学奥林匹克竞赛金牌教练,全国青少年信息学奥林匹克竞赛十大杰出指导教师
- 陈合力** 绍兴市第一中学教师,国际信息学奥林匹克竞赛金牌教练,全国青少年信息学奥林匹克竞赛十大杰出指导教师
- 胡旭红** 杭州第二中学教师,国际信息学奥林匹克竞赛金牌教练,全国青少年信息学奥林匹克竞赛优秀指导教师
- 邱桂香** 东北育才学校教师,国际信息学奥林匹克竞赛金牌教练,全国青少年信息学奥林匹克竞赛十大杰出指导教师
- 杨慧蓉** 乌鲁木齐市第一中学教师,国际信息学奥林匹克竞赛金牌教练
- 游光辉** 绍兴市第一中学教师,国际信息学奥林匹克竞赛金牌教练,全国青少年信息学奥林匹克竞赛优秀指导教师

## 丛书主审

- 朱全民** 长沙市雅礼中学教师,国际信息学奥林匹克竞赛金牌教练,全国青少年信息学奥林匹克竞赛十大杰出指导教师

# 前 言

本书以 Pascal 语言为载体,重点介绍分析问题、算法设计和编写程序的方法和技巧。计算机语言初学者经常出现没有深入分析问题就直接编写程序的现象,导致出现自己无法解决的错误,从而影响学习的积极性。本书采用与其他同类 Pascal 语言教材不同的书写手法,把问题分析和算法设计摆在第一位置,仅把计算机语言当成描述和实现算法的工具。

本书每章由三大部分组成:

(1)问题导引与分析。试图通过这一部分先引发学生思考,用已有的知识提出解决问题的方案,在实施方案过程中,发现问题,进而带着问题学习第二部分的语言知识内容。

(2)Pascal 语言知识。为每章需要学习的 Pascal 语言语句结构,通过应用例子和解决问题导引的例子,让学生掌握语句的灵活运用。

(3)程序设计实例。为章节知识和算法相结合部分,通过有一定难度的问题实例,让学生综合应用本章的知识,通过问题分析、算法描述和程序设计三个步骤提升学生的综合思维能力。

本书除了完整地介绍 Pascal 语言程序设计外,还穿插经典基础算法的分析和介绍,以螺旋上升方式呈现语言知识和算法设计有机结合,尽量通过经典基础算法实例展开讨论,让学生学会从不同的角度、用不同的方法分析和解决问题,培养学生的思维能力。如:第 6 章数组,当学生学完数组之后,从程序设计角度可以解决很多的问题了,因此,在第 6 章中加入了大量的经典基础算法,让学生利用现有的语言知识,从问题分析、算法设计到程序实现做一个提升,使学生的学习注意力能够集中在如何提高分析问题和解决问题的能力上。当思维能力提升了,接下去再继续学习语言知识就比较容易了。本书注意让学生在程序设计的學習过程中,培养算法的效率意识,逐渐提高有效的算法设计能力。

本书的最后一章综合程序设计,以奥赛试题方式展开一些算法讨论,试图通过这一章的学习,让学生了解更多的算法知识和分析实现技巧,全面提升基础算法设计与综合程序设计能力。更希望学习者通过这一章的学习能够产生

问号而不是句号,带着更多的问号和学习的欲望进行更深入的学习和研究。

本书读者对象为中学生、大学生、信息学教师、信息学奥赛教练和程序设计爱好者。学生学完本书将有能力参加信息学奥林匹克竞赛。

本书第 1、2、5、6、9 章和第 11 章的 3、4 节由福州一中的陈颖老师编写。第 3、4、7、8、10 章和第 11 章的 1、2 节由东北育才学校的邱桂香老师编写。

在本书搜集资料和调试程序的过程中,福州一中信息组谢路昱、林承宇、王昌翰、刘冬煜、胡张广达、谢易非等同学和东北育才学校王正宇、解东同学,复旦大学的张旭,上海交通大学的李诗剑同学,做了大量的工作。在此,向他们表示感谢。

作者

2010 年 4 月

# 目 录

<b>第 1 章 计算机和计算机语言</b> .....	1
1.1 计算机工作原理 .....	1
1.1.1 计算机基本组成 .....	1
1.1.2 计算机工作原理 .....	2
1.1.3 计算机软件 .....	3
1.2 计算机语言 .....	3
1.2.1 计算机语言的发展进程 .....	3
1.2.2 程序设计进程 .....	4
1.2.3 Pascal 语言概述 .....	6
1.3 算法和算法描述 .....	10
1.3.1 程序设计的基本方法 .....	10
1.3.2 算法 .....	11
1.3.3 算法的描述 .....	12
1.4 练习题 .....	16
<b>第 2 章 顺序结构程序设计</b> .....	19
2.1 问题导引与分析 .....	19
2.1.1 问题导引 .....	19
2.1.2 问题分析 .....	19
2.1.3 解决方案 .....	21
2.2 变量、常量定义 .....	21
2.2.1 变量的物理意义 .....	21
2.2.2 变量的定义 .....	22
2.2.3 常量的定义 .....	23
2.3 赋值语句 .....	24
2.3.1 赋值语句的格式和功能 .....	24
2.3.2 Pascal 常用函数库 .....	25
2.3.3 表达式规范书写格式 .....	26
2.3.4 赋值语句的应用 .....	28

2.4	输出语句 .....	29
2.4.1	输出语句的格式和功能 .....	29
2.4.2	写语句的输出格式 .....	30
2.4.3	输出语句的应用 .....	31
2.5	输入语句 .....	32
2.5.1	输入语句的格式与功能 .....	32
2.5.2	read 和 readln 语句区别 .....	33
2.5.3	输入语句的应用 .....	34
2.6	顺序结构程序设计实例 .....	35
2.6.1	实例 2-1 日期写法(date) .....	35
2.6.2	实例 2-2 数字分离(splitnum) .....	36
2.6.3	实例 2-3 时间戳(times) .....	38
2.7	练习题 .....	39
<b>第 3 章 选择结构程序设计 .....</b>		<b>42</b>
3.1	问题导引与分析 .....	42
3.1.1	问题导引 .....	42
3.1.2	问题分析 .....	42
3.1.3	解决方案 .....	43
3.2	布尔表达式 .....	44
3.3	条件语句 .....	45
3.3.1	条件语句的格式与功能 .....	45
3.3.2	条件语句的应用 .....	47
3.4	情况语句 .....	52
3.4.1	情况语句的格式与功能 .....	52
3.4.2	情况语句的应用 .....	54
3.5	选择结构程序设计实例 .....	58
3.5.1	实例 3-1 机器人的移动问题(robot) .....	58
3.5.2	实例 3-2 解方程的困扰(equation) .....	60
3.5.3	实例 3-3 数表读数(table) .....	63
3.6	练习题 .....	66
<b>第 4 章 循环结构程序设计 .....</b>		<b>70</b>
4.1	问题导引与分析 .....	70
4.1.1	问题导引 .....	70
4.1.2	问题分析 .....	71
4.1.3	解决方案 .....	72

4.2	for 循环语句 .....	72
4.2.1	for 循环语句的格式与功能 .....	72
4.2.2	for 循环语句的应用 .....	75
4.3	while 循环语句 .....	82
4.3.1	while 循环语句的格式与功能 .....	82
4.3.2	while 循环语句的应用 .....	83
4.3.3	while 循环语句与 for 循环语句应用的互换 .....	86
4.4	repeat 循环语句 .....	88
4.4.1	repeat 循环语句的格式与功能 .....	88
4.4.2	repeat 循环语句的应用 .....	89
4.4.3	repeat 循环语句与 while 循环语句的互换 .....	93
4.5	循环结构程序设计实例 .....	96
4.5.1	实例 4-1 出租车计费(taxicost) .....	96
4.5.2	实例 4-2 价格竞猜(price) .....	98
4.5.3	实例 4-3 满载着爱的代码(heart) .....	99
4.6	练习题 .....	101
<b>第 5 章 文 件 .....</b>		<b>105</b>
5.1	问题导引与分析 .....	105
5.1.1	问题导引 .....	105
5.1.2	问题分析 .....	107
5.1.3	解决方案 .....	108
5.2	文本文件 .....	108
5.2.1	文本文件的特点 .....	109
5.2.2	文本文件的操作 .....	109
5.3	文件应用实例 .....	112
5.3.1	实例 5-1 笑笑和“最佳运动员”问题——创建测试数据文件举例 .....	112
5.3.2	实例 5-2 乐乐和“乒乓球”问题——创建测试数据文件举例 .....	114
5.3.3	实例 5-3 文件合并——多文件数据操作举例 .....	116
5.4	练习题 .....	117
<b>第 6 章 数 组 .....</b>		<b>122</b>
6.1	问题导引与分析 .....	122
6.1.1	问题导引 .....	122
6.1.2	问题分析 .....	124

6.1.3	解决方案	126
6.2	一维数组	126
6.2.1	一维数组的定义	126
6.2.2	一维数组的引用	127
6.2.3	一维数组的应用	129
6.3	多维数组	140
6.3.1	多维数组的定义	140
6.3.2	二维数组的应用	141
6.4	数组应用实例	146
6.4.1	实例 6-1 矩形(maxt)——枚举法应用举例	146
6.4.2	实例 6-2 火柴棒等式(matches)——枚举法应用举例	150
6.4.3	实例 6-3 最大子段和(seq)——贪心思想应用举例	153
6.4.4	实例 6-4 集合删数(number)——贪心思想应用举例	155
6.4.5	实例 6-5 奇因数(fsum)——递推算法应用举例	159
6.4.6	实例 6-6 守望者的逃离(escape)——递推算法应用举例	161
6.5	练习题	164

## 第 7 章 函数与过程 171

7.1	问题导引与分析	171
7.1.1	问题导引	171
7.1.2	问题分析	174
7.1.3	解决方案	175
7.2	函数	177
7.2.1	函数的定义	177
7.2.2	函数的调用	179
7.2.3	函数的应用	179
7.3	过程	183
7.3.1	过程的定义	184
7.3.2	过程的调用	184
7.3.3	过程的应用	185
7.4	函数与过程的数据传递	189
7.4.1	数值参数与变量参数	189
7.4.2	全局变量与局部变量	191
7.5	函数与过程的嵌套	193
7.5.1	函数与过程的嵌套	193
7.5.2	递归程序设计	194
7.6	函数与过程程序设计实例	204

7.6.1	实例 7-1 一元三次方程求解(equation)——分治法应用举例	204
7.6.2	实例 7-2 八皇后问题(queen)——回溯法应用举例	206
7.6.3	实例 7-3 积木覆盖(puzzle)——回溯法应用举例	208
7.7	练习题	211
<b>第 8 章 字符串</b>		218
8.1	问题导引与分析	218
8.1.1	问题导引	218
8.1.2	问题分析	220
8.1.3	解决方案	221
8.2	字符串	221
8.2.1	字符串的定义	221
8.2.2	字符串的基本操作	223
8.2.3	字符串函数和过程	224
8.2.4	字符串的应用	228
8.3	字符串综合应用程序设计实例	231
8.3.1	实例 8-1 ISBN 号码(ISBN)——模拟法应用举例	231
8.3.2	实例 8-2 连接整数(link)——贪心法应用举例	233
8.3.3	实例 8-3 单词接龙(words)——回溯法应用举例	234
8.4	练习题	237
<b>第 9 章 记录、集合、枚举及子界类型</b>		243
9.1	问题导引与分析	243
9.1.1	问题导引	243
9.1.2	问题分析	246
9.1.3	解决方案	247
9.2	记录类型	247
9.2.1	记录类型定义	248
9.2.2	记录类型的操作	250
9.2.3	记录类型的应用	252
9.3	集合类型	255
9.3.1	集合类型的定义	255
9.3.2	集合的构成	256
9.3.3	集合的运算	256
9.3.4	集合的应用	258
9.4	枚举类型	259

9.4.1	枚举类型的定义	260
9.4.2	枚举类型的性质	260
9.4.3	枚举类型的应用	261
9.5	子界类型	263
9.5.1	子界类型的定义	263
9.5.2	子界类型的应用	264
9.6	自定义数据类型程序设计实例	265
9.6.1	实例 9-1 作业调度方案问题(jsp)——记录类型应用举例	265
9.6.2	实例 9-2 整数因子团问题(ise)——集合类型应用举例	270
9.7	练习题	272
<b>第 10 章</b>	<b>指针类型</b>	<b>277</b>
10.1	问题导引与分析	277
10.1.1	问题导引	277
10.1.2	问题分析	278
10.1.3	解决方案	279
10.2	指针及其应用	280
10.2.1	指针变量	280
10.2.2	动态变量	281
10.2.3	指针的操作	282
10.3	链表结构	283
10.3.1	链表的基本结构	283
10.3.2	单向链表的基本操作	285
10.3.3	其他链表结构	290
10.4	指针类型程序设计实例	291
10.4.1	实例 10-1 合并珠子(merge)——链表的合并操作举例	291
10.4.2	实例 10-2 约瑟夫问题(joseph)——链环的应用举例	294
10.4.3	实例 10-3 走出泥潭(route)——应用链表求解路径问题举例	296
10.5	练习题	299
<b>第 11 章</b>	<b>综合程序设计</b>	<b>306</b>
11.1	贪心算法专题	306
11.1.1	贪心算法思想	308
11.1.2	购买贺年卡——生活中的贪心算法举例	309
11.1.3	设置喷水池——覆盖问题中的贪心算法举例	311

11.1.4	筛选游戏卡——数学问题中的贪心算法举例	313
11.1.5	舞伴的搭配——匹配问题中的贪心算法举例	314
11.2	递推算法专题	316
11.2.1	递推算法思想	318
11.2.2	淘气的钥匙——错排问题中的递推算法举例	319
11.2.3	生日蛋糕——几何问题中的递推算法举例	320
11.2.4	交费金额——存在性问题中的递推算法举例	321
11.2.5	谁去谁留——概率统计问题中的递推算法举例	322
11.3	高精度专题	324
11.3.1	高精度算法思想	326
11.3.2	奖励——高精度加法和减法举例	327
11.3.3	月赛——高精度乘法和除法举例	329
11.3.4	求 $2^p$ ——高精度乘法中的分治运用举例	331
11.3.5	二叉树计数——Catalan 数举例	333
11.4	基础搜索专题	336
11.4.1	基础搜索算法思想	339
11.4.2	迷宫问题——深度优先和广度优先搜索举例	342
11.4.3	零和问题——搜索参数的选择举例	344
11.4.4	01 距离——搜索对象的选择举例	347
11.4.5	小木棍——搜索剪枝举例	349
	参考文献	353

# 计算机和计算机语言

在今天,计算机对大家而言已不是陌生的设备,人们可以使用同一台计算机做不同的事,这是计算机和其他设备不同的重要特色之一。计算机之所以能执行不同的工作任务,是基于其硬件和软件协同工作的工作机制。

要使计算机按人们指定的步骤有效地工作,必须先编制好一组让计算机执行的指令,这就是程序。随着计算机软件技术的发展,人们可以通过不同的计算机语言来编写程序。

进行程序设计,需要人们对问题进行分析,确定了解决问题的方法和步骤,然后根据它选择计算机语言编写出程序,再让计算机执行这个程序,获得问题的结果。

学习程序设计要掌握并灵活运用计算机语言,更要研究解决问题的分析方法和有效的解决步骤,即“算法”。

通过本章的学习,将使读者了解计算机的工作原理,了解计算机语言、程序设计和算法的基本概念,为后面章节的 Pascal 语言程序设计学习奠定思维基础。

## 1.1 计算机工作原理

### 1.1.1 计算机基本组成

自 1946 年第一台计算机(ENIAC)诞生之后,计算机硬件技术与计算机应用技术飞速发展,形成了以计算机为核心的数字信息时代。

计算机的基本组成如图 1.1 所示。

计算机硬件电路规定了计算机只能接受由 0、1 组成的二进制数字信息,即计算机硬件中流动的信息均为二进制信息。

中央处理器,简称 CPU,由运算器和控制器组成,是计算机的核心部件。运算器主要完成各种算术运算和逻辑运算。运算器一次能运算多少位二进制数及在单位时间内能做多少次运算是计算机工作速度的重要指标。控制器的作用是分析程序指令,控制计算机各部件按程序指令协调地工作。

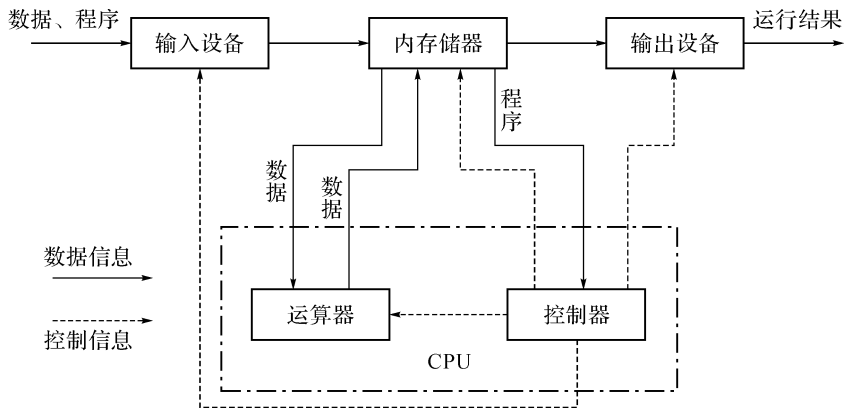


图 1.1 计算机的基本组成

内存储器是计算机的记忆部件,用于存放程序指令的数据。计算机内存储器如同一个大仓库,包含许多存储单元,为了便于管理,每个存储单元都有一个编号称为地址。存储器有一个特点,从某一存储单元中“取”一个数据后,该单元中的数据并不消失,除非向该单元送入一个新的数据后,此单元的内容才改变,因此把数据的“存”和“取”称为“写入”和“读出”。

输入、输出设备是计算机与外界交换信息的设备,通过 I/O 接口与计算机的主板连接。输入设备的作用是将信息输入计算机内存储器,常见的输入设备有:键盘、鼠标、扫描仪等。输出设备是将计算机内的信息输出,常见的输出设备有:显示器、打印机、音箱等。

硬盘、U 盘、软盘等通常称为计算机的外存储器,属于输入、输出设备。

位(Bit)是计算机存储器容量的最小单位,也是度量信息的最小单位。存储器容量常用单位有:字节(Byte)、千字节(KB)、兆字节(MB)、千兆字节(GB)。

它们之间的换算关系如下:1Byte = 8Bit; 1KB = 1024Byte; 1MB = 1024KB; 1GB = 1024MB。

### 1.1.2 计算机工作原理

计算机的工作机制是硬件的任何操作受指令控制,指令的集合即为程序,计算机工作流程为:

- (1)通过输入设备将原始数据和程序存入存储器中;
- (2)控制器从存储器中获取程序指令,按照指令发出控制信息流,指挥计算机各部件协调工作;
- (3)需要运算时,存储器将数据传输至运算器,运算器运算后结果传回至存储器;
- (4)需要输出时,存储器将结果数据传输至输出设备实现输出。

程序与数据在计算机内部都是以二进制形式表示,这种“以二进制数的形式存储程序与数据,按顺序执行程序中的指令”的原理,是由匈牙利美籍科学家冯·诺依曼(Von Neumann, 1903—1957年)提出来的,称之为冯·诺依曼原理,也就是存储程序原理。

### 1.1.3 计算机软件

根据计算机工作原理,要使计算机有条不紊地工作,光有计算机硬件还不够,还必须要有计算机软件,两者配合共同完成信息处理的任务。

计算机硬件本身只能完成一些很简单的基本操作,如加法、减法、传送数据、收发控制电压脉冲等,这些简单的基本操作叫做计算机指令。一台计算机有不止几十条指令,把它们合起来叫计算机的指令系统。计算机无论做多么复杂和高级的工作,都是通过逐条地执行一个由指令组合的指令序列,最后完成整个工作。这种指令排列成一定的执行顺序并能完成一定目标功能的指令序列,就叫做程序。

软件就是一大段程序,具有专门而完善的功能。例如,Word 字处理软件有完善的文字编辑功能。软件开发属于程序设计范畴。

软件包括系统软件和应用软件。为了方便用户使用机器及其输入输出设备,充分发挥计算机系统的效率,围绕计算机系统本身开发的程序系统叫做系统软件。如我们使用的操作系统(常用的有 Windows、Unix 等)、语言编译程序、数据库管理软件系统等。应用软件是专门为了某种使用目的而编写的程序系统,常用的有文字处理软件、办公软件、专用的财务软件、人事管理软件、计算机辅助软件、绘图软件、学校学生成绩管理软件等。软件具有知识产权且受到保护。

## 1.2 计算机语言

### 1.2.1 计算机语言的发展进程

要使计算机能按人的规定完成一系列的工作,首先要解决一个“语言”沟通问题,在人和计算机之间构建一种两者都能识别的特定的语言,使计算机具备理解并执行人们给出的各种指令的能力,这种特定的语言称之为计算机语言,也叫程序设计语言,它是人和计算机沟通的桥梁。

随着计算机技术的迅速发展,程序设计语言经历了由低级向高级发展的多个阶段,程序设计方法也得到了不断地发展和提高。

计算机语言按其发展程度可以划分为:机器语言、汇编语言和高级语言。

#### 1. 机器语言

计算机并不能理解和执行人们使用的自然语言,计算机硬件能够直接识别的指令是由一连串的 0 和 1 组合起来的二进制编码,称为机器指令,每一条指令规定了计算机要完成的某个操作。机器语言则是指计算机能够直接识别的指令的集合,它是最早出现的计算机语言。

例如,图 1.2 是某一种型号计算机的一组二进制编码机器指令,用来完成一个简单的加法运算:9+8。

这一组二进制编码的计算机指令序列构成了一个简单的机器语言程序。显然,用机器语言编写的程序使人“难学、难记、难写、难

```
10110000
00001001
00000100
00001000
11110100
```

图 1.2 机器指令

检查、难调试”,给使用者带来很大的不便。机器语言编写程序的另一个缺点是:完全依赖于机器硬件,不同型号的机器其机器语言指令不相同,程序的可移植性差。其优点是计算机能直接识别、执行效率高。

### 2. 汇编语言

为了克服机器语言的缺点,人们对机器语言进行了改进,用一些容易记忆和辨别的有意义的符号代替机器指令。用这样一些符号代替机器指令所产生的语言就称为汇编语言,也称为符号语言。用汇编语言来实现  $9+8$  运算的有关指令如图 1.3。

MOV AL, 9  
ADD AL, 8

图 1.3 汇编语言

### 3. 高级语言

汇编语言虽然较机器语言有所改善,但未从根本上摆脱指令系统的束缚,它与机器语言指令仍然是一一对应的,而且与自然语言相距甚远,不符合人们的表达习惯。

为了从根本上改变语言体系,必须从两方面下功夫:一是力求接近于自然语言;二是力求脱离具体机器,使语言与机器指令系统无关,达到程序通用的目的。20 世纪 50 年代中期出现了第一个高级语言——Fortran 语言,随之相继出现了许多种高级语言。

高级语言与具体的计算机无关,即用它所写的程序可以在不同机型的计算机上运行,高级语言较之汇编语言更接近于自然语言。描述计算公式与数学上的表示大体一致,例如,前面计算  $9+8$  的问题,若用 Basic、C 语言或 Visual Basic 语言编程,只要用  $S=9+8$  一个语句就可完成,十分简单,而且易于理解。

显然,计算机不能直接执行高级语言程序,而必须先将程序翻译转换成机器能够识别的“目标程序”才能执行,这种翻译转换工作由被称为“编译程序”的专门软件来完成,每一种高级语言都有自己的编译程序。在一台计算机上运行某一种高级语言源程序的前提是:该计算机安装了该语言的编译程序。

高级语言亦称为面向过程的算法语言,用它编写程序必须写出每一步如何进行的全过程,程序设计者必须具体指出执行的每一个细节。有些场合人们进一步希望能做到,只要指出“做什么”而不必具体指出“如何做”,由计算机自己去解决“如何做”的问题,这就是非过程化语言,在 20 世纪 70 年代后期,研制出了非过程化语言,例如 Forth 语言、dBASE 等数据库语言,显然,非过程化语言的出现为更多的人方便地使用计算机创造了极有利的条件。

不同的语言体系适于解决不同类别的问题,大多数计算机上运行的程序,多是用高级语言编写的,高级语言程序设计是目前学习计算机知识的一项基本训练。

## 1.2.2 程序设计进程

高级语言出现后,使得更多人有能力开发计算机应用软件。早期的程序盲目追求效率,不注意程序的可读性,程序设计无章可循,导致 20 世纪 60 年代一批大型软件相继开发失败,出现了所谓的“软件危机”。

在 20 世纪 60 年代末出现了结构化程序设计,强调从程序结构和程序风格上来研究程序开发,使程序易于编写和调试,易于阅读和维护。

结构化程序设计是面向问题处理的过程,它主要包括以下三方面的内容: