



“十一五”高等院校精品规划教材

“SHI YI WU” GAODENG YUANXIAO JINGPIN GUIHUA JIAOCAI

XINBIAN
FORTRAN 90
CHENGXU SHEJI
JIAOCHENG

新编 **FORTRAN 90** 程序设计教程

○ 编著 白云 李学哲 贾波



北京交通大学出版社
<http://press.bjtu.edu.cn>

“十一五”高等院校精品规划教材

新编 FORTRAN 90 程序设计教程

编 著 白 云 李学哲 贾 波
主 审 钱培德

北京交通大学出版社

• 北京 •

内 容 简 介

本书共分 16 章, 主要内容有: 程序设计概述、FORTRAN 90 软件开发环境、FORTRAN 90 基础知识、内部数据类型与顺序结构程序设计、数据有格式输入输出、选择结构程序设计、循环结构程序设计、数组、函数与子例行程序、派生类型与结构体、指针与动态数据结构、文件与设备、接口与模块、公用区和存储关联、图形处理、Visual Basic 与 FORTRAN 90 的多语言混合编程。本书可用作高等院校程序设计课教材、各类计算机培训教材和工程技术人员参考用书。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

新编 FORTRAN 90 程序设计教程/白云, 李学哲, 贾波编著. —北京: 北京交通大学出版社, 2009. 1

(“十一五”高等院校精品规划教材)

ISBN 978-7-81123-266-0

I. 新… II. ①白… ②李… ③贾… III. FORTRAN 语言-程序设计-高等学校-教材
IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 049182 号

责任编辑: 史鸿飞

出版发行: 北京交通大学出版社

电话: 010-51686414

北京市海淀区高粱桥斜街 44 号

邮编: 100044

印刷者: 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: 26.25 字数: 650 千字

版 次: 2009 年 2 月第 1 版 2009 年 2 月第 1 次印刷

书 号: ISBN 978-7-81123-266-0/TP·410

印 数: 1~3 000 册 定价: 39.80 元

本书如有质量问题, 请向北京交通大学出版社质监组反映。对您的意见和批评, 我们表示欢迎和感谢。

投诉电话: 010-51686043, 51686008; 传真: 010-62225406; E-mail: press@bjtu.edu.cn。

前 言

学习和掌握程序设计语言及程序设计技术对提高计算机应用能力具有重要作用,也有助于提高综合素质和创新能力,这也是当今社会对大学生和工程技术人员的基本要求。只有熟练掌握程序设计语言及程序设计技术,才能更好地运用计算机求解各种复杂问题。

FORTRAN 语言是最早出现的高级语言,它的诞生对程序设计语言的研究和发展产生了极其深刻而广泛的影响。长期以来,FORTRAN 语言被广泛应用于数值分析、工程设计和科学计算等领域,并以严谨、规范、高效的特有功用受到人们的普遍青睐。目前,数目庞大且遍布全球的 FORTRAN 程序成为人类一笔非常宝贵的物质财富,为社会进步和经济发展做出了贡献。

FORTRAN 语言具有顽强的生命力。在 50 多年的发展过程中,它的功能不断增强,版本不断更新,影响不断扩大,历经岁月,经久不衰。在面向对象和可视化程序设计技术风靡全球,以及 Visual C++、Visual Basic、Visual FoxPro 等可视化编程语言盛行天下的今天,FORTRAN 语言不甘落后,积极引入先进思想和理念,紧跟程序设计语言前进的步伐,于 1991 年发布和推出了功能强大、特色鲜明的 FORTRAN 90,使传统的 FORTRAN 语言焕发出了现代气息和新的活力。

美国微软公司 (MS) 在可视化编程工具的研究和开发领域称雄天下,数据设备公司 (DEC) 在高质量 FORTRAN 编译器和科学计算方面独树一帜,两者强强联合,优势互补,合作研究、开发和推出了功能强大的 Digital Visual FORTRAN 5.0,使 Microsoft Developer Studio 可视化集成开发环境与优质 FORTRAN 编译器得到完美结合。Digital Visual FORTRAN 5.0 的推出,使 FORTRAN 90 锦上添花,更胜一筹,在社会上产生了广泛影响,受到人们的普遍欢迎。

本书通俗易懂,循序渐进,由浅入深,突出概念,注重方法,强化编程,并配以简洁说明、丰富实例和详细题解,为读者提供全面、详尽及实用的 FORTRAN 90 程序设计内容。每章后附有部分习题,供读者平时练习或上机实验之用。本书有配套的电子教学课件和实验指导书,需要者可与作者联系。E-mail: by59@163.com 或 Liguopingsh@163.com。

苏州大学钱培德教授负责本书主审工作,提出了许多宝贵的修改意见;河海大学王志坚教授和南京大学陈华生教授也对书稿进行了仔细的审阅,并提出了许多建设性的修改意见,保证了本书的水平和质量,在此表示诚挚的谢意。

本书由白云、李学哲和贾波编著,刘怡、刘敏、陈国新、高洁羽、黄兰兰和陶均煜等同志参加了本书的部分编写工作或对教材的编写提供了帮助,在此一并表示感谢。白云负责全书策划、审阅和定稿工作。

由于编者水平有限,书中难免存在不妥之处,恳请读者批评指正。

编 者

2009 年 1 月

目 录

第 1 章 程序设计概述	1
1.1 程序设计语言	1
1.1.1 程序设计语言分类	1
1.1.2 程序设计语言概念	3
1.1.3 程序设计语言形式化描述	3
1.2 程序和程序设计	4
1.2.1 程序	4
1.2.2 程序设计	5
1.2.3 程序设计步骤	5
1.3 算法的基本概念和特征	6
1.3.1 算法含义	6
1.3.2 算法特征	6
1.3.3 算法评价标准	6
1.3.4 算法描述	7
1.4 程序设计方法	11
1.4.1 模块化程序设计方法	11
1.4.2 结构化程序设计方法	11
1.4.3 面向对象程序设计方法	12
习题 1	12
第 2 章 FORTRAN 90 软件开发环境	14
2.1 FORTRAN 语言发展概述	14
2.2 FORTRAN 90 的主要特征	14
2.3 Digital Visual FORTRAN 5.0 的主要特性	16
2.4 Digital Visual FORTRAN 5.0 软硬件配置和安装	17
2.5 Microsoft Developer Studio 开发环境概述	18
2.5.1 Microsoft Developer Studio 简介	18
2.5.2 工作空间 (Workspace) 和项目 (Project)	18
2.5.3 启动 Microsoft Developer Studio	19
2.5.4 Microsoft Developer Studio 主窗口说明	19
2.6 Microsoft Developer Studio 解题过程	25
2.6.1 前期准备工作	25

2.6.2	创建工作空间	26
2.6.3	创建项目	27
2.6.4	创建源程序文件	28
2.6.5	创建辅助文档文件	29
2.6.6	编译源程序文件	31
2.6.7	构建可执行程序文件	32
2.6.8	运行程序	33
2.6.9	调试程序	34
2.6.10	程序多区域显示	37
习题 2		38
第 3 章	FORTRAN 90 基础知识	40
3.1	字符集	40
3.2	名称	40
3.3	关键字	41
3.4	程序单元	41
3.5	语句	43
3.6	程序书写格式	44
3.7	数据类型	45
习题 3		47
第 4 章	内部数据类型与顺序结构程序设计	49
4.1	常量	50
4.1.1	整型常量	50
4.1.2	实型常量	51
4.1.3	复型常量	54
4.1.4	字符型常量	55
4.1.5	逻辑型常量	56
4.2	变量	56
4.2.1	变量名称	56
4.2.2	变量类型	57
4.2.3	整型变量	58
4.2.4	实型变量	59
4.2.5	复型变量	60
4.2.6	字符型变量	61
4.2.7	逻辑型变量	62
4.3	表达式	63
4.3.1	算术表达式	63
4.3.2	字符表达式	65

4.3.3 关系表达式	66
4.3.4 逻辑表达式	67
4.3.5 表达式运算的误差问题及其解决办法	69
4.4 赋值语句	70
4.4.1 算术赋值语句	70
4.4.2 字符赋值语句	71
4.4.3 逻辑赋值语句	72
4.5 表控输入输出语句	72
4.5.1 表控输入语句	73
4.5.2 表控输出语句	75
4.6 参数语句 (PARAMETER 语句)	78
4.7 函数	79
4.8 END 语句、STOP 语句和 PAUSE 语句	81
4.8.1 END 语句	81
4.8.2 STOP 语句	82
4.8.3 PAUSE 语句	82
4.9 简单顺序程序设计举例	82
习题 4	85
第 5 章 数据有格式输入输出	89
5.1 概述	89
5.2 格式说明与格式编辑符	89
5.3 整数有格式输入输出	92
5.4 实数有格式输入输出	94
5.4.1 小数型实数有格式输入输出	94
5.4.2 指数型实数有格式输入输出	97
5.4.3 双精度型实数有格式输入输出	99
5.5 复数有格式输入输出	99
5.6 字符串有格式输入输出	100
5.6.1 A 编辑符	100
5.6.2 字符串编辑符	102
5.7 逻辑值有格式输入输出	102
5.8 二、八、十六进制数据有格式输入输出	103
5.8.1 数据的二进制编码表示	103
5.8.2 二、八、十六进制数据有格式输入输出	106
5.9 不可重复编辑符	108
5.9.1 X 编辑符	108
5.9.2 / 编辑符	109

5.9.3 \ 编辑符	110
5.9.4 T、TL、TR 编辑符	111
5.9.5 : 编辑符	111
5.9.6 Q 编辑符	112
5.10 纵向走纸控制	112
5.11 输入输出表与格式说明表的相互关系	114
5.12 有格式输入输出应用举例	117
习题 5	118
第 6 章 选择结构程序设计	122
6.1 概述	122
6.2 逻辑 IF 语句	123
6.3 块 IF 语句	124
6.3.1 基本块 IF 语句	124
6.3.2 多分支块 IF 语句	127
6.3.3 嵌套的块 IF 语句	131
6.3.4 块 CASE 选择结构	132
6.4 选择结构程序设计举例	134
习题 6	137
第 7 章 循环结构程序设计	139
7.1 循环结构概述	139
7.1.1 循环及循环结构	139
7.1.2 循环结构分类	141
7.2 用 DO 循环语句实现“计数型”循环	143
7.2.1 DO 循环语句	143
7.2.2 DO 循环语句的执行过程	145
7.3 用 DO WHILE 循环语句实现循环	147
7.3.1 DO WHILE 循环语句	147
7.3.2 DO WHILE 循环语句的执行过程	148
7.4 强制性终止循环过程 (EXIT、CYCLE)	149
7.5 特殊 DO 循环语句与 EXIT 语句	150
7.6 循环嵌套	151
7.7 循环结构程序设计举例	153
习题 7	160
第 8 章 数组	162
8.1 概述	162
8.2 数组声明	164
8.2.1 用类型声明语句进行数组声明	164

8.2.2 用 DIMENSION 语句进行数组声明	166
8.2.3 用类型声明语句和 DIMENSION 属性进行数组声明	166
8.3 数组元素引用	167
8.4 数组的逻辑结构和存储结构	170
8.5 数组的输入输出	173
8.5.1 使用 DO 循环语句输入输出数组	173
8.5.2 使用隐含 DO 循环子句输入输出数组	175
8.5.3 使用数组名或数组片段输入输出数组	177
8.6 给数组赋初值	178
8.6.1 通过数组构造器给数组赋初值	178
8.6.2 通过 DATA 语句给数组赋初值	179
8.7 WHERE 语句	180
8.7.1 逻辑 WHERE 语句	180
8.7.2 块 WHERE 语句	181
8.7.3 多分支 WHERE 语句	182
8.8 动态数组	184
8.9 数组应用举例	187
习题 8	200
第 9 章 函数与子例行程序	203
9.1 概述	203
9.2 标准子程序	209
9.2.1 标准函数	209
9.2.2 标准子例行程序	210
9.3 语句函数	212
9.4 内部子程序	217
9.4.1 内部函数子程序	217
9.4.2 内部子例行程序	223
9.5 子程序形参与实参之间的数据传递关系	227
9.5.1 变量作形式参数	228
9.5.2 数组作形式参数	229
9.5.3 可调数组名作形式参数	232
9.5.4 假定大小数组作形式参数	234
9.5.5 假定形状数组作形式参数	235
9.5.6 子程序名作形式参数	236
9.5.7 星号作形式参数	241
9.5.8 形式参数与实在参数的结合关系	242
9.6 递归子程序	243

9.6.1 递归的定义	243
9.6.2 递归函数	243
9.6.3 递归子例行程序	244
9.7 外部子程序	245
习题 9	251
第 10 章 派生类型与结构体	253
10.1 概述	253
10.2 派生类型定义	254
10.3 结构体声明	256
10.4 结构体成员引用	257
10.5 结构体的初始化	258
10.6 结构体数据的输入和输出	259
10.6.1 结构体数据输入	259
10.6.2 结构体数据输出	260
10.7 派生类型应用举例	261
习题 10	265
第 11 章 指针与动态数据结构	267
11.1 概述	267
11.1.1 指针的存储结构和访问方式	267
11.1.2 指针声明	269
11.1.3 指针状态	270
11.2 指针引用和赋值	270
11.2.1 指针引用	270
11.2.2 指针赋值	271
11.2.3 结构体与指针	272
11.3 整型指针	273
11.4 与指针相关的函数和语句	274
11.4.1 置空语句: NULLIFY	274
11.4.2 链接测试函数: ASSOCIATED	275
11.4.3 动态存储分配语句: ALLOCATE 和 DEALLOCATE	275
11.5 指针数组	276
11.5.1 指针数组	276
11.5.2 指针数组作为函数返回值	277
11.6 动态链表	278
11.6.1 链表概念	278
11.6.2 创建和搜索链表	278
11.6.3 插入结点	280

11.6.4 删除结点	282
11.6.5 环形和双向链表	284
习题 11	285
第 12 章 文件与设备	286
12.1 概述	286
12.2 物理设备和逻辑设备	288
12.3 文件	289
12.3.1 外部文件	290
12.3.2 内部文件	291
12.3.3 文件标识	292
12.4 外部文件分类	292
12.4.1 文件存储格式	292
12.4.2 文件存取方式	293
12.4.3 文件定位	295
12.5 文件基本操作	295
12.5.1 文件打开 (OPEN)	295
12.5.2 文件关闭 (CLOSE)	299
12.5.3 文件结束 (ENDFILE)	300
12.5.4 文件输入 (READ)	300
12.5.5 文件输出 (WRITE)	304
12.5.6 文件查询 (INQUIRE)	304
12.5.7 文件指针反绕 (REWIND)	308
12.5.8 文件指针回退 (BACKSPACE)	309
12.6 使用外部设备	309
12.7 文件的读写操作	310
12.7.1 有格式顺序存取文件读写操作	310
12.7.2 有格式直接存取文件读写操作	311
12.7.3 无格式顺序存取文件读写操作	313
12.7.4 无格式直接存取文件读写操作	314
12.7.5 二进制顺序存取文件读写操作	315
12.7.6 二进制直接存取文件读写操作	316
12.8 文件应用举例	317
习题 12	321
第 13 章 接口与模块	322
13.1 接口界面块	322
13.2 模块	324
13.3 重载	329

13.3.1 子程序重载	329
13.3.2 操作符重载	331
13.3.3 赋值号重载	335
习题 13	336
第 14 章 公用区和存储关联	338
14.1 公用区	338
14.1.1 COMMON 语句	339
14.1.2 公用区特性和使用规则	340
14.1.3 有名公用区的作用	342
14.2 存储关联 (等价)	343
14.2.1 EQUIVALENCE 语句	344
14.2.2 EQUIVALENCE 语句使用规则	344
14.3 数据块子程序	346
习题 14	348
第 15 章 图形处理	350
15.1 图形处理概述	350
15.1.1 坐标系统	350
15.1.2 图形颜色	354
15.1.3 线型	355
15.1.4 填充样式	356
15.2 QuickWin 绘图设计基础	357
15.3 设置 QuickWin 窗口	360
15.4 多窗口操作	362
15.5 常用绘图函数	366
15.5.1 绘制像素点	366
15.5.2 设置当前绘图点	367
15.5.3 绘制直线	367
15.5.4 绘制形状	368
15.6 文字信息的显示	374
15.7 图形应用实例	377
习题 15	389
第 16 章 Visual Basic 与 FORTRAN 90 的多语言混合编程	390
16.1 概述	390
16.1.1 动态链接库	390
16.1.2 动态链接库的优点	391
16.1.3 动态链接库的实现方法	391
16.2 Visual Basic 与 FORTRAN 90 混合编程	391

16.2.1 Visual Basic 与 FORTRAN 90 混合编程基础	391
16.2.2 Visual Basic 与 FORTRAN 90 混合编程步骤	394
16.3 Visual Basic 与 FORTRAN 90 混合编程应用	395
习题 16	404
参考文献	405

第 1 章 程序设计概述

对于计算机系统,如果说 CPU 是其心脏,那么程序就是其灵魂。学习和掌握程序设计是驾驭和运用计算机系统的重要一环。本章主要介绍有关程序设计的一些基本知识、基本概念和基本方法,为学习和掌握 FORTRAN 90 程序设计奠定基础。

1.1 程序设计语言

计算机系统是在“程序”的控制下自动进行工作的,人们通过程序设计语言编写的程序来指挥和控制计算机运行。程序设计语言是人与计算机进行交流的重要工具,随计算机诞生而诞生,伴随计算机发展而发展,在计算机科学技术的发展过程中,发挥了巨大作用。

1.1.1 程序设计语言分类

从计算机诞生之日起,程序设计语言从低级语言到高级语言,从传统语言到现代语言不断向前发展,新的、功能强大的程序设计语言不断涌现。目前,程序设计语言有上百种。

程序设计语言属于人工语言、形式语言、非自然语言。分为低级语言和高级语言两种。

低级语言又分为机器语言和汇编语言。机器语言是一种计算机能直接理解、识别和执行的程序设计语言,它由 0 和 1 两个二进制符号按照确定的规则描述和表达。例如,计算 $2+3$,并从屏幕上输出,用机器语言编写的程序为

0010001100000010	-----将 2 送累加器 A
0000001100000011	-----将 3 与累加器相加,结果 5 在累加器
00000010	-----将累加器数 5 从总线输出

机器语言与具体的计算机硬件有关,不同种类计算机配置有不同的机器语言,如同不同国家、不同民族有不同的自然语言一样。对于计算机来说,识别、理解和执行机器语言极为容易,对人类来说却是苦不堪言,只有极少数非常专业的计算机专家才能使用机器语言编写程序,绝大多数人不用机器语言编写程序,但是了解一点机器语言知识也有益处。机器语言的最大优点是运行效率极高,在一些嵌入式控制系统中会用到机器语言程序。机器语言的缺点是难以理解、阅读及编写程序,程序的可靠性、可维护性及可移植性差。

为解决机器语言存在的突出问题,克服其不足和缺点,出现了汇编语言(或称符号语言)。汇编语言是用一些易于理解的符号来取代机器语言中难于理解的二进制编码,如用“ADD”代替加法的二进制编码“00000011”。用汇编语言替代机器语言编写程序,可提高程序的可读性、可靠性和可维护性。例如,计算 $2+3$,并从屏幕上输出,用汇编语言编写的程序为

MOV A,2H	-----将 2 送累加器 A
ADD A,3H	-----将 3 与累加器相加,结果 5 在累加器
OUTL BUS,A	-----将累加器数 5 从总线输出



汇编语言同机器语言一样，与计算机硬件有关，不同种类的计算机配置有不同的汇编语言。汇编语言的优点是执行速度快，占用空间少，运行效率高，其缺点是程序不易编写、理解、维护和移植，一般情况下不使用汇编语言，但是在一些嵌入式控制系统中还习惯采用其编写程序，以提高运行效率，所以汇编语言还有其生存空间。

汇编语言程序必须翻译转换成机器语言程序后，才能被计算机识别、理解和执行，翻译转换工作由一个称为“汇编程序”的软件来自动完成。翻译过程称为“汇编”，如图 1-1 所示，被翻译的汇编语言程序称为“源程序”，翻译后的机器语言程序称为“目标程序”。

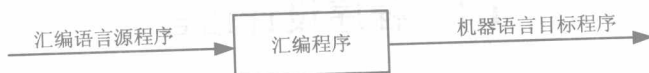


图 1-1 汇编过程

高级语言是其表达方式接近自然语言和数学语言的一类程序设计语言。高级语言克服了低级语言的固有缺点，它面向过程、易读易学、易于编写、可靠性高、可维护性好，基本上与计算机硬件无关，能在不同配置的计算机上运行。例如，计算 $2+3$ ，并从屏幕上输出，用高级语言编写程序只需用一条语句“PRINT *,2+3”即可。对于一个复杂问题，若用高级语言编写程序来解决，其优点更加明显。如计算物体运动距离的 FORTRAN 90 语言程序为

```

! 计算物体运动距离
PROGRAM calculate
REAL v0,a,t,s
READ *,v0,a,t !输入初速度、加速度和运动时间
s=v0*t+0.5*a*t*t
PRINT *, 's=',s
END
  
```

用高级语言便于描述解题算法，故高级语言也称“算法语言”。高级语言的出现开创了计算机科学技术发展的新纪元，是计算机发展史上的一次革命。FORTRAN 语言是最早出现的高级语言，受其影响，之后高级语言如雨后春笋般不断涌现，目前高级语言有上百种，但常用的有十多种。不同的高级语言其词法、语法和语义不尽相同，应用范围和领域也存在差异。

高级语言可分为过程式语言、函数式语言、逻辑语言、结构化语言、模块化语言等，有的高级语言具有多种特征。取消落后内容，引入先进思想，反映现代特征，成为高级语言发展的主要趋势。FORTRAN 90 就是传统 FORTRAN 语言焕发出现代气息的一种高级语言。

高级语言程序不能直接被计算机识别、理解 and 执行，必须将其翻译转换成机器语言程序，或者先翻译转换成汇编语言程序，再通过汇编程序翻译转换成机器语言程序，翻译转换工作由称为“编译程序”的软件自动完成。翻译过程称为“编译”，如图 1-2 所示，被翻译的高级语言程序称为“源程序”，编译后的机器语言程序称为“目标程序”。

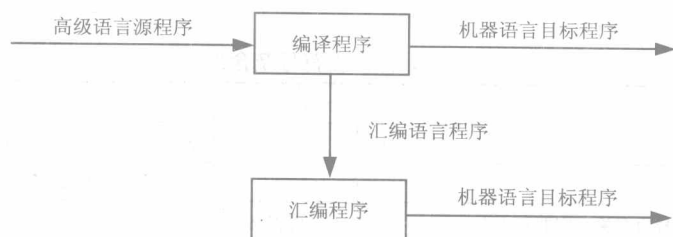


图 1-2 编译过程

1.1.2 程序设计语言概念

程序设计语言种类繁多，形式各异，功能有别。它们同自然语言一样都有严格的词法、语法和语义规定，不同之处是程序设计语言不允许出现二义性、歧义性和不确定性。词法、语法和语义概念是程序设计语言中的 3 个最基本概念，学习程序设计语言的一个重要内容就是深入理解和全面掌握其词法、语法和语义规定。

(1) 词法。组成合法语句的基本单位称为单词，构成合法单词的规则称为词法。词法通常采用严格的形式化描述，也有采用语法图描述。程序设计语言规定了一组描述词法的规则，称为词法规则。通过词法规则可判断一个单词是否合法，含有非法单词的程序在计算机上不能运行，程序必须符合语言的词法规则。如语句“PRINT *,sum”中的“PRINT”、“*”、“,”、“sum”都是符合 FORTRAN 语言词法规则的合法单词，如果将“PRINT”改写成“PRN”，由于“PRN”不是一个合法单词，所以修改后语句就不能在计算机上运行。

(2) 语法。具有特定含义的符号串或句子称为语句，构成合法语句的规则称为语法。语法通常采用形式化描述，也有采用语法图描述。程序设计语言规定了一组描述语法的规则。通过语法规则可判断一个语句是否合法，含有非法语句的程序在计算机上不能运行，程序必须符合语言的语法规则。如语句“PRINT *,sum”就是符合语法规则的合法语句，如果将其改写成“PRINT +,sum”就不是一个合法语句。虽然“PRINT”、“+”、“,”、“sum”都是 FORTRAN 语言的合法单词，但“PRINT”后直接跟一个“+”号，不符合 PRINT 语句的语法规则，所以不是一个合法语句，不能在计算机上运行。

(3) 语义。合法语句的实际含义称为该语句的语义。语义通常采用自然语言描述。程序设计语言规定了每一个语句的语义。对语义的理解是否准确，直接影响程序的可靠性和正确性。语义理解有误，程序就可能蕴涵错误，准确理解语句的语义，对于程序设计来说至关重要。若将输出语句“PRINT *,sum”的语义理解为输入 sum 值，则程序运行结果就不可能正确。

1.1.3 程序设计语言形式化描述

程序设计语言的词法和语法一般采用形式化描述，语义一般采用自然语言描述。形式化描述手段有很多，BNF（即巴科斯范式）是最常用的一种描述手段，可看作是描述程序设计语言的形式化语言，它定义了一组描述语言的规则。通常称 BNF 为元语言，称程序设计语言为对象语言，称规则为产生式，称规则中不属于对象语言的符号为元语言符号（或



连接词), 称属于对象语言的符号为终结符, 称定义产生式的符号为非终结符。BNF 中元语言符号如表 1-1 所示。

表 1-1 元语言符号及含义

元语言符号	含 义	元语言符号	含 义
→	定义为	{ }	表示括号内对象可出现任意次, 也可不出现
	或者	< >	表示括号内为元语言符号, 非终结符
()	表示括号内对象一定取其中之一	"" 可省略	表示引号内为对象语言符号, 终结符
[]	表示括号内任取一个对象, 也可不取		

例如, 采用 BNF 描述 FORTRAN 语言“标识符”的一组产生式规则为

<标识符> → <字母> { <字母> | <数字> | "_" }

<字母> → "大写英文字母" | "小写英文字母"

<数字> → "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

这组规则严格而精确地定义了 FORTRAN 语言中的标识符, 其含义是: 标识符由若干个字符组成, 第一个字符必须是英文字母, 其后可以是英文字母、数字或下划线。这组规则没有规定标识符长度, 具体长度由具体计算机系统或编译程序规定, 需查阅编译系统用户手册。由此可看出 BNF 描述比自然语言描述简洁、规范、清晰、精确和严格, 本书采用 BNF 描述。

描述词法和语法有时也用图形(即语法图)描述, 如描述标识符的语法图如图 1-3 所示。

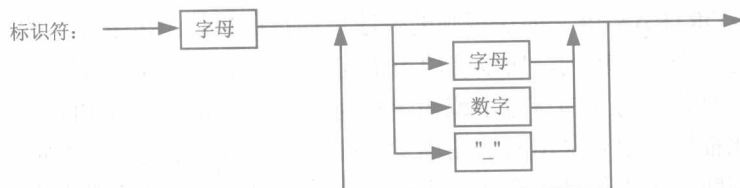


图 1-3 描述标识符的语法图

1.2 程序和程序设计

对于程序设计语言, 不但要学习其词法、语法和语义, 更重要的是学习如何灵活使用这些规则来求解实际问题, 学习编写程序的方法, 让计算机按照人们的意愿进行工作。

1.2.1 程序

程序设计语言是全部计算机指令(语句)的集合, 程序是符合程序设计语言词法、语法和语义规则的计算机指令(语句)序列, 语句序列规定了语句的执行顺序。用程序设计语言可写出许多不同的程序, 计算机的神奇之处就在于可以执行许多奥妙无穷的程序。

下面是一个计算任意圆柱体体积的 FORTRAN 90 程序。

! 已知圆柱体的半径 r 和高度 h, 求它的体积 V, 并输出

```
READ *,r,h
```

```
V=3.1415*r*r*h
```