

Ada 程序设计导论

张丽芬 编著



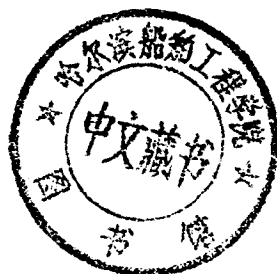
北京理工大学出版社

77312
218

363411

Ada 程序设计导论

张丽芬 编著



北京理工大学出版社

(京)新登字 149 号

内 容 提 要

Ada 语言是一种功能很强的计算机高级语言。它不仅具有一般算法语言的数值计算和数据处理功能,而且为实时任务控制和实时信息处理提供了强有力的设施。它适应于各种规模的计算机。用于嵌入式计算机时,可实现对各种目标的控制。

本书是按教材风格编写,由浅入深,易于掌握。全书共分十章,介绍 Ada 的基本概念、组成及其特点,并列举了大量的工程应用实例。每章后附有习题。读者不仅可以从书中学会新的程序设计语言,而且可以学会一种新的程序设计方法。

本书可作为大专院校的计算机、信息工程、电子通讯等专业二年级以上学生和研究生的必修和选修教材,也可供计算机科学和应用领域的科技工作者参考使用。

Ada 程序设计导论

张丽芬 编著

※

北京理工大学出版社出版发行

各地新华书店经售

国防科工委印刷厂印刷

※

850×1168 毫米 32 开本 9.75 印张 255 千字

1992 年 10 月第一版 1992 年 10 月第一次印刷

ISBN 7-81013-552-X/TP·52

印数:1-4000 册 定价:5.50 元

前 言

Ada 作为一种功能很强的新颖的计算机高级语言,已经受到高度重视,成为 80 年代最有影响和代表性的通用语言。

Ada 是美国国防部为克服软件危机在 1979 年研制成功的。它不仅能进行一般的数值计算和数据处理,更重要的是为实时控制和实时信息处理提供了强有力的设施。它适用于各种规模的计算机,尤其适用于嵌入式计算机,实现对目标的控制。与其它语言相比,它提供了许多新的概念,如程序包的封闭性,数据的抽象表示,并行操作的任务等。因此,Ada 的出现在许多方面推动了程序设计语言技术的发展。

基于上述原因,Ada 在我国也已受到广泛重视,并被定为我军的标准程序设计语言。为了适应这种形势,北京理工大学计算机系从 1988 年起就在计算机专业的本科生中开设 Ada 课程,讲授 Ada 程序设计。本书就是为满足教学需要而编写的。初稿完成于 1987 年,根据教学中发现的问题,每年均有所修改、补充。这次交出版社正式出版前,编者又进一步作了加工,力求既能满足高等学校有关专业教学的需要,又能满足社会各方面了解和学习 Ada 语言的渴望。

本书共分十章。第一章至第三章介绍 Ada 语言的基本组成和基本概念。第四章至第十章介绍 Ada 语言区别于其它语言所具有的特色。每章都有实例和习题,以帮助读者掌握概念,加深理解。本书中所举例子,有的来自作者的科研成果,这些例子不仅在 VAX 机上而且在微机上顺利通过编译和运行。实例涉及的面很广,有常用的数值计算、数据处理,又有实时控制和并发任务的实现等,学

习本书以后,读者可以举一反三,用 Ada 语言编写程序,解决工作中遇到的各种实际问题

本书由北京理工大学吴鹤龄教授主审,审阅中提出了许多宝贵意见,在此表示衷心感谢。

编者

1991 年 12 月

目 录

第一章 引论

1—1 一个简单的 Ada 程序	(2)
1—2 Ada 基本说明	(4)
一、Ada 的字符集	(4)
二、Ada 的词法单元	(5)
三、数值直接量	(5)
四、字符直接量和字符串	(6)
五、限制符	(7)
1—3 类型和目标说明	(7)
一、类型说明	(7)
二、目标说明	(8)
1—4 标量类型	(9)
一、整型	(9)
二、枚举类型	(9)
三、离散类型的属性	(10)
四、字符型和布尔型	(11)
五、实型	(11)
1—5 组合数据类型	(14)
一、数组	(15)
二、记录	(20)
1—6 表达式和操作符说明	(23)
一、表达式之例	(24)
二、操作符的说明	(24)
1—7 语句	(27)
一、赋值语句	(27)
二、空语句	(27)

三、条件语句	(27)
四、case 选择语句	(28)
五、循环语句	(29)
六、出口语句	(31)
七、无条件转移语句	(32)
八、返回语句	(33)
九、块结构	(33)
习题	(34)

第二章 子程序

2—1 子程序组成	(36)
一、子程序说明	(36)
二、子程序体	(37)
三、形式参数	(38)
四、子程序举例	(39)
2—2 子程序调用	(41)
一、实现对数据交换子程序的调用	(41)
二、实现对函数 SQRT 的调用	(42)
三、实现对函数重载操作符的调用	(43)
2—3 子程序重载	(44)
2—4 可见性规则	(45)
一、实体的作用域	(45)
二、实体的可见性	(46)
2—5 重命名实体	(49)
一、程序包的换名	(49)
二、子程序的换名	(50)
2—6 举例	(50)
一、八皇后问题	(50)
二、求分类数组中的最频值	(55)
三、汉诺(HANOI)塔问题	(57)
四、数组的快速分类算法	(58)
五、矩阵乘法	(60)
习题	(62)

第三章 输入和输出

3—1 引言	(63)
3—2 直接文件和顺序文件	(64)
一、对直接和顺序文件的存取	(64)
二、对直接和顺序文件的操作命令	(66)
三、文件操作中的异常	(68)
四、举例	(69)
3—3 正文文件	(70)
一、缺省文件的输入输出	(71)
二、控制打印格式	(72)
三、对正文文件的存取	(77)
四、交互式输入	(78)
五、正文文件输入输出示例	(78)
习题	(82)

第四章 对类型的进一步说明

4—1 关于强类型	(84)
4—2 子类型	(84)
4—3 派生类型	(86)
4—4 类型转换	(87)
4—5 存取类型	(88)
一、存取类型说明	(88)
二、变量的动态分配	(89)
三、动态分配变量的命名	(91)
四、存取类型的作用	(92)
五、存取类型和索引限制	(98)
4—6 记录类型	(99)
一、带有判别式的记录类型	(99)
二、变体记录	(101)
4—7 举例	(103)
一、用变体记录类型计算几何图形的面积	(103)
二、打印二叉树的值	(104)
习题	(105)

第五章 程序包

5—1 引言	(107)
5—2 程序包说明	(107)
5—3 程序包体	(109)
5—4 私有类型	(111)
5—5 举例	(120)
一、符号表管理程序包	(120)
二、优先级队列管理程序包	(125)
习题	(129)

第六章 独立编译

6—1 引言	(131)
6—2 程序库	(132)
6—3 编译单元和编译	(133)
6—4 子单元(SUBUNIT)	(134)
6—5 编译顺序	(142)
6—6 重新编译	(143)
习题	(143)

第七章 异常处理

7—1 引言	(148)
7—2 异常说明和异常处理	(149)
一、异常说明	(149)
二、异常处理程序	(149)
7—3 系统预定义的异常	(152)
7—4 引发异常的语句	(153)
7—5 异常处理程序的激活和异常传播	(154)
一、在处理说明期间引发异常	(154)
二、在语句执行期间引发异常	(154)
三、异常处理举例	(155)
7—6 再现引发异常的操作	(156)
7—7 抑制异常检查	(157)
7—8 异常处理举例	(161)
一、阶乘	(161)

二、化简公式	(161)
三、文件合并	(162)
习题	(165)
第八章 任务	
8—1 引言	(167)
8—2 任务和会合	(168)
8—3 任务描述与任务体	(169)
8—4 接收入口调用语句	(170)
8—5 任务类型和任务目标	(174)
一、任务类型	(174)
二、动态分配的任务	(175)
8—6 任务的激活与终止	(176)
一、任务的激活	(176)
二、任务的终止	(177)
8—7 延时语句和停止语句	(178)
一、延时语句	(178)
二、停止语句	(178)
8—8 选择语句	(178)
一、选择等待语句	(179)
二、条件入口调用语句	(181)
三、定时入口调用语句	(181)
8—9 互斥	(183)
8—10 任务的入口族	(184)
8—11 异常结束语句	(186)
8—12 中断	(187)
8—13 任务优先级	(187)
8—14 任务和入口的属性	(188)
8—15 任务和异常	(188)
8—16 举例	(189)
一、用 Ada 任务实现信号	(189)
二、信号灯	(190)
三、任务调度控制	(191)

四、最短作业优先调度	(193)
五、哲学家进餐问题	(194)
六、电梯控制	(198)
七、核反应堆控制	(210)
习题	(213)

第九章 通用单元

9—1 引言	(215)
9—2 通用单元说明和体	(217)
9—3 通用形参说明及匹配规则	(220)
一、通用类型的参数	(220)
二、通用子程序参数	(221)
9—4 举例	(223)
一、求指定元素的第 N 个后继元素	(223)
二、堆栈	(224)
三、通用插入分类过程	(225)
四、梯形积分法	(227)
五、集合	(229)
习题	(231)

第十章 低级程序设计

10—1 引言	(234)
10—2 系统特性的改变	(236)
10—3 表示属性和表示子句	(237)
一、表示属性	(237)
二、表示子句	(239)
10—4 与其它语言接口	(248)
一、编用 INTERFACE	(248)
二、输入输出编用	(250)
三、插入机器代码	(255)
10—5 不检查地回收存贮区	(256)
10—6 不经检查的类型转换	(256)
10—7 举例	(257)
一、打字机的输入输出	(258)

二、存贮分配程序	(261)
习题	(263)
附录 1 预定义的语言属性	(265)
附录 2 预定义的语言编用	(272)
附录 3 预定义的语言环境	(278)
附录 4 预定义的程序包 CALENDAR	(292)
附录 5 VAX 机上的 SYSTEM 程序包	(294)
参考文献	(299)

第一章 引 论

Ada 是高级语言不断发展的产物,其历史可追溯到 1974 年。当时软件在计算机系统成本中所占的比例已经上升到 30%以上(目前这个比例数字就更大了),对于嵌入式系统,软件消耗尤其庞大。所谓嵌入式计算机,就是配置在更大的电子机械装置上的那些实时计算机。例如,大型的武器系统、飞机、战舰、空间飞行器、快速传输网络、报文交换系统等。这样一些复杂系统中所使用的计算机就是嵌入式计算机。嵌入式计算机软件具有如下特征:

- ①要求运行速度快;
- ②要求程序紧凑;
- ③程序量大(5 万到 10 万行);
- ④生存周期长(10 到 15 年);
- ⑤维护人员流动快(一般为 2 年);
- ⑥不仅输出数据,而且要输出控制信号;
- ⑦程序、数据需要经常修改。

针对以上要求,人们常用汇编语言或专用语言来编写这类软件。根据美国国防部(DOD)的统计,当时至少有 450 种语言用于它的各种嵌入式系统,使嵌入式系统软件的开发与维护极为困难,费用急剧增加。为此,DOD 决心谋求语言的标准化,以寻求出路。1975 年,美国国防部成立了一个由陆、海、空三军代表组成的高级语言工作组(HOLWG),最后选定了 Ada 语言并正式列入 ANSI 标准。

尽管开始时 Ada 是作为美国国防部嵌入式系统的专用语言而研制的,但最终它成为一种通用语言,既适合于军用,又适合于

民用,既适用于嵌入式系统,又适用于非嵌入式系统。它是一种模块化语言,来源于 pascal,但比 pascal 更严密,功能更强。它不仅具有控制抽象和过程抽象,而且还具有数据抽象。

1—1 一个简单的 Ada 程序

为了说明 Ada 语言的基本结构和特点,先引入一个模拟计算器加、减、乘、除计算的 Ada 程序:

```
with TEXT_IO; use TEXT_IO;
procedure CALCULATOR is
    type REAL is digits 10;
    package IO_REAL is new FLOAT_IO(REAL);
    use IO_REAL;
    A,B      : REAL;
    OPR      : CHARACTER;
    RESULT : REAL;
begin
    while not END_OF_FILE(STANDARD_INPUT) loop
        GET(A);GET(OPR);GET(B);
        Case OPR is
            when '+' => RESULT := A+B;
            when '-' => RESULT := A-B;
            when '*' => RESULT := A*B;
            when '/' => RESULT := A/B;
            when others=>
                PUT("ERROR * * * BAD OPERATOR");
                exit;
            end case;
        PUT(RESULT);NEW_LINE;
    end loop;
end CALCULATOR;
```

这是一个 Ada 的主程序,名为 CALCULATOR。程序名在程序开始(Procedure 引导)和结束(end 引导)处都要给出,从而使程序体明显地定界。

程序的第一行(with TEXT_IO; use TEXT_IO;)调用系统定义的程序包 TEXT_IO,以实现输入、输出。with 子句指明本程序需要使用包含正文输入输出功能的程序包 TEXT_IO; use 子句允许本程序在调用 TEXT_IO 中的子程序时,无须明确指出 TEXT_IO 这个程序包的前缀。TEXT_IO 不仅为一些预定义的数据类型提供实现输入输出的子程序,而且也为其一些类型(如 FLOAT_IO)提供输入输出子程序。例如,本程序第四行:

```
package IO_REAL is new FLOAT_IO(REAL);
```

就是为浮点类型 REAL 生成输入输出子程序的程序包。

在 Ada 中,procedure、is、begin、end 等是程序的主要结构。with、use 用来为程序运行提供所需要的预先定义的程序包。这些词都是 Ada 的关键字。必须用小写字母表示。

Ada 允许一行中包含多个语句,但每一个语句必须以分号结尾。为了提高程序的可读性和作为文档的需要,可在程序中的任何地方插入注释。注释以两个短横线“--”开始,一直延续到该行的结束。以上示例程序加上必要的注释以后是这样的:

```
with TEXT_IO; use TEXT_IO;
--使输入输出能实现
procedure CALCULATOR is
    type REAL is digits 10;
    --实数的精度至少为 10 位数字
    package IO_REAL is new FLOAT_IO(REAL);
    use IO_REAL; --现在可进行实数输入输出
    A,B      :REAL; --A,B 定义为实型变量
    OPR      :CHARACTER;
    RESULT:REAL;
begin
```

```

while not END_OF_FILE(STANDARD_INPUT) loop
    -- 函数 STANDARD_INPUT 返回一个缺省输入文件
    GET(A);GET(OPR);GET(B);-- 读输入
    case OPR is -- case 语句
        when '+' => RESULT := A+B;
        when '-' => RESULT := A-B;
        when '*' => RESULT := A*B;
        when '/' => RESULT := A/B;
        when others =>
            PUT("ERROR * * * BAD OPERATOR");
            exit; -- 当输入错误时,退出循环
    end case;
    PUT(RESULT);NEW_LINE;-- 写输出
end loop;
end CALCULATOR;

```

1-2 Ada 基本说明

一、Ada 的字符集

Ada 的基本字符包括 26 个大写字母、10 个阿拉伯数字、一个空格字符和下面给出的一些特殊字符,共 56 个。

" , # , & , ' , (,) , + , - , * , / , . , ' , : , ; , < , = , > , - , |

除了基本字符外,Ada 的扩充字符包括 26 个小写字母和下面一些特殊字符,共 39 个。

! , \$, % , ? , @ , [, \ ,] , ^ , ` , { , } , ~

任何 Ada 程序都可以只用基本字符集中的字符编写而成。任何包括扩充字符在内的 Ada 程序,也都可以转换成只包含基本字符的等价的 Ada 程序。程序中的小写字母除用在字符串和字符直接量中的以外,都等价于相应的大写字母。

二、Ada 的词法单元

Ada 的词法单元包括标识符、数值直接量、字符直接量、字符串、限制符以及注释。

标识符由字母打头,后跟任意个字母、数字,也可以嵌入下横线以增加可读性。相邻的词法单元可由任意多个空格分隔,但一个词法单元必须放在同一行上。

有些标识符也看成是 Ada 保留字,如 TEXT_IO。不计这样保留字,Ada 共有 63 个保留字,保留字在程序中以小写形式出现,如表 1—1 所示。

表 1—1 Ada 保留字

			of	
	declare		or	select
abort	delay	generic	others	separate
abs	delta	goto	out	subtype
accept	digits	if	package	task
access	do	in	pragma	terminate
all	else	is	private	then
and	elsif	limited	procedure	type
array	end	loop	raise	use
at	entry	mod	range	when
begin	exception	new	record	while
body	exit	not	renames	with
case	for	null	return	xor
constant	function		reverse	

三、数值直接量

数值直接量有两种形式——整数和实数。数值可以用十进制表示,也可以用二、八、十六进制表示。为了便于阅读,在相邻的数