

程序设计语言Ada

刘炳文 编著

TP312

国防工业出版社

程序设计语言 Ada

刘炳文 编著

国防工业出版社

(京)新登字 106 号

内 容 简 介

Ada 是大型通用程序设计语言,是现代计算机语言的成功代表。它是迄今为止最完备的软件工具,能在整个软件生存期中支持软件工程各项目标的实现,从而可有效地降低软件费用。此外,Ada 还为软件系统的集成、开发和工程化提供了一种强有力的工具。

本书以 ANSI/MIL-STD-1815A 为基础,深入浅出地讲述了 Ada 的全部设施,通过大量实例,系统而全面地介绍了 Ada 语言的程序设计方法。书中所有程序都可以在 IBM-PC 及其兼容机上编译、连接、运行。全书在编排上注意了循序渐进,由浅入深,由简及繁,理论与实践密切结合。在每章后附有适当的练习题。

本书既可作为高等院校有关专业、计算机学习班的教材,也可作为计算机软件工作者和计算机应用人员自学的参考书。

JS46-137
02

程序设计语言 Ada

刘炳文 编著

责任编辑 鹿啟 炳

*

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

新华书店经售

北京市怀柔县王史山印刷厂印刷

787×1092 毫米 16 开本 印张 29 $\frac{1}{4}$ 682 千字

1993 年 9 月第一版 1993 年 9 月第一次印刷 印数:0001-3000 册

ISBN 7-118-01010-3/TP·127 定价:24.40 元

前 言

Ada 是美国国防部为克服软件危机,耗费巨资,历时近 10 年,在国际范围内发起研制的一种崭新的程序设计语言。1980 年 7 月正式发表了 Ada 语言的标准文本,并被定为美国军用标准(MIL/STD-1815A)。此后,Ada 语言文本几经修订,于 1983 年被定为美国国家标准(ANSI1815A),1987 年 3 月,国际标准化组织接受 Ada 语言为国际标准语言。

Ada 是迄今为止功能最完备的一种程序设计语言,被誉为第四代语言的成功代表。它具有结构清晰、修改方便、模块化能力强、所编写的程序可重复使用等特点,而且在减少程序的研制费用和增加可移植性方面,它比任何一种语言都优越。因此,Ada 语言的使用,可大大改善软件系统的清晰性、可靠性、有效性和可维护性。同时,Ada 还具有其他语言所不具备的许多功能(诸如并发控制、类属程序单元、异常处理、数据抽象、信息隐藏、分别编译等),并支持面向对象的程序设计方法。此外,Ada 还是软件工程的基础。专家们预测,20 世纪 90 年代到 21 世纪初,Ada 将是占统治地位的语言。

从 80 年代中期开始,我国开展了对 Ada 语言及其编程环境的研究工作,目前在 Ada 编译程序、Ada 工具、环境、语言转换、交叉编译、Ada 方法学基础研究等方面已经取得了一定的成就,具备了推广应用 Ada 的条件。为了适应国际上迅速发展的形势,在我国推广和普及 Ada 语言,笔者在总结前段 Ada 教学实践经验和体会的基础上,参考了国内外有关 Ada 方面的专著,编写了本书。该书以标准 Ada(ANSI/MIL-STD-1815A)为背景,以 Ada 的初学者为对象,深入浅出地讲述了在微型机上用 Ada 语言进行程序设计的方法。全书共 16 章,系统地介绍了 Ada 的词法元素、流程控制、各种数据类型、子程序、程序包、类属、任务、异常以及输入/输出等设施。在编排上,本书注意了由简及繁、由浅入深、循序渐进、理论与实践的密切结合,注重在程序设计方面的实用性,力图使读者在较短的时间内,能通过本书掌握 Ada 程序设计方法。

Ada 语言功能完备,但不像其他语言那样普及,再加上宣传多,使用少,因而蒙上了一层神秘的色彩,致使很多人在它面前望而却步,不敢问津。本书充分注意到这种情况,既考虑了科技人员的需要,又照顾到初学者的接受能力,深入浅出地介绍了 Ada 的全部设施。根据大多数初学者的特点,本书在讨论 Ada 程序设计的过程中,尽量从实际问题入手,逐步引出概念,然后阐述理论,得出结论。在编写具体的程序时,先讲清目的、思路,指出初学者易犯的错误,使读者养成正确地用 Ada 语言进行程序设计的良好习惯。

Ada 是第四代语言,不依赖于具体的机器和操作系统。但是,如果只“纸上谈兵”,不编写能在具体机器上运行的程序,要掌握 Ada 程序设计技术是不可能的。为此,本书选择了业经美国国防部确认、可以在微型机上运行的 Meridian Ada Vantage 编译系统进行程序设计。该系统完全符合 Ada 标准,已由有关部门引进,并进行了汉化,不仅能用于教学,而且可以用来解决实际问题,这对推广和普及 Ada 将起到决定性的作用。本书就是以国内

主流机型 IBM-PC 和 Meridian Ada Vantage 编译系统为基础编写而成的。书中凡涉及与实现有关的特征时,均以此编译系统为准。书末的附录 1 简单介绍了该编译系统的使用方法。本书的所有例子都给出了完整的程序,这些程序都已在微型机上运行通过。

考虑到某些微型机上没有安装 8087 或 80287 数学协处理器,无法进行浮点运算,本书中的大部分程序是针对整型数的操作编写的。但这对编程方法和算法没有任何影响。

《程序设计语言 Ada》注重实用,紧密结合程序设计实际。其着眼点不是介绍语言文本,而是力图使读者真正掌握 Ada 程序设计技术,并能用来解决实际问题。

在编写本书的过程中,得到了空军指挥学院陈材保教授、林彬副教授的大力协助。陈材保教授还为本书提供了部分例题,在此深表谢意。

Ada 是一种大型的、全新的程序设计语言,在我国的应用刚开始起步。笔者希望本书能成为引玉之砖,激发读者对学习和使用 Ada 语言的兴趣,加速在我国推广和普及 Ada 的进程。但是,由于水平所限,不妥之处在所难免,恳请广大读者批评指正。

刘炳文

1993 年 1 月于国防大学

目 录

第一章 概论	1
1.1 为什么要学习 Ada 语言	1
1.2 Ada 的特点	3
1.3 国外对 Ada 的开发、应用情况	5
1.4 Ada 的构成	7
习题	7
第二章 Ada 词法元素	8
2.1 字符集	8
2.2 标识符	8
2.3 数值字面量	10
2.4 字符和字符串	11
2.5 注释	12
习题	12
第三章 标量数据类型	14
3.1 Ada 的类型分类与机理	14
3.2 整数类型	18
3.3 实数据类型	20
3.4 枚举数据类型	22
3.5 类型转换	26
习题	28
第四章 简单程序设计	30
4.1 Ada 语言的基本运算及表达式	30
4.2 Ada 程序的结构	34
4.3 赋值语句	37
4.4 正文输出	39
4.5 正文输入	47
4.6 空语句	48
4.7 简单程序举例	49
习题	51
第五章 流程控制	53
5.1 条件语句	53
5.2 循环语句	61
5.3 情况语句	77
5.4 转移语句	80
5.5 块语句	81
5.6 应用举例	83

习题	92
第六章 子程序	95
6.1 简单的子程序	95
6.2 子程序声明	97
6.3 子程序调用	103
6.4 过程子程序	106
6.5 函数子程序	110
6.6 全程变量和局部变量	114
6.7 子程序嵌套	115
6.8 递归	120
6.9 重载	124
6.10 应用举例	129
习题	134
第七章 程序包	136
7.1 引例	136
7.2 程序包的语法	139
7.3 程序包的使用	145
7.4 use 子句	150
7.5 声明的重命名	153
7.6 无体程序包	155
7.7 程序包与私有类型	157
7.8 派生类型与程序包	160
7.9 预定义程序包	161
习题	162
第八章 数组类型	164
8.1 数组类型的定义和对象声明	164
8.2 数组的聚集	167
8.3 数组的操作	169
8.4 数组作为子程序参数	177
8.5 多维数组	181
8.6 字符串	184
8.7 数组类型转换	186
习题	186
第九章 记录类型	188
9.1 记录类型的定义	189
9.2 记录聚集	192
9.3 记录的操作	192
9.4 应用举例	202
习题	211
第十章 判别类型	213
10.1 可变记录	213
10.2 判别式约束	216

10.3	变长数组	218
10.4	判别类型的使用	219
10.5	判别私有类型	224
	习题	225
第十一章 高级输入/输出		227
11.1	文件	227
11.2	文件的声明	229
11.3	文件的打开与关闭	234
11.4	顺序文件	236
11.5	随机存取文件	240
11.6	正文文件	242
11.7	数值数据文件的输入/输出	253
11.8	字符串输入/输出	256
11.9	高级输入输出中的异常处理	257
11.10	应用举例	261
	习题	270
第十二章 访问数据类型		272
12.1	基本概念	272
12.2	访问类型声明	274
12.3	动态分配	276
12.4	动态分配变量的命名	279
12.5	访问类型的操作	282
12.6	链表	287
12.7	双向链表	291
12.8	树	293
12.9	判别类型作为基类型	298
12.10	应用举例	299
	习题	304
第十三章 分别编译		306
13.1	引言	306
13.2	程序库	307
13.3	编译单元与 with 子句	308
13.4	子单元	314
13.5	编译的顺序	316
13.6	重新编译	317
13.7	作用域和可见性	318
	习题	319
第十四章 类属程序单元		321
14.1	引例	321
14.2	模板——类属声明	324
14.3	类属例化	331
14.4	类属形式对象和类属形式子程序	335

14.5 类属形式类型	341
习题	349
第十五章 异常	352
15.1 引例	352
15.2 异常的引发	354
15.3 异常的传播	359
15.4 预定义异常	361
15.5 程序员定义的异常	364
15.6 异常应用举例	367
习题	371
第十六章 任务	372
16.1 任务类型与任务对象	373
16.2 任务的并发执行	377
16.3 简单会合	380
16.4 入口语句和接收语句	386
16.5 更为实用的会合	390
16.6 更为灵活的会合	404
16.7 任务的高级特性	407
16.8 任务的属性	411
16.9 任务与异常	412
16.10 应用举例	416
习题	426
附录	428
附录 1 Meridian Ada Vantage 编译系统使用简介	428
附录 2 预定义语言环境	431
附录 3 预定义语言属性	437
附录 4 预定义语言编用	442
附录 5 预定义标识符	445
附录 6 Ada 语法规则	447
语法索引	456
主要参考文献	460

第一章 概 论

Ada 是一种表现能力很强的通用程序设计语言,它是美国国防部为克服软件开发危机,耗费巨资、历时近 10 年研制成功的。它被誉为第四代计算机语言的成功代表。与其他流行的程序设计语言不同,它不仅体现了许多现代软件的开发原理,而且将这些原理付诸实现。因此,Ada 语言的使用可大大改善软件系统的清晰性、可靠性、有效性和可维护性。

美国国防部之所以把这种语言取名为 Ada,是为了纪念奥左斯特·艾达·洛夫莱斯伯爵夫人(Augusta Ada Lovlace,1815~1852 年),她是英格兰诗人拜伦(Byron)勋爵的女儿,曾对现代计算机技术之父查尔斯·巴贝奇(Charles Babage)的笔记、手稿进行了整理和修正。从某种意义上说,她是世界上第一位计算机程序员。

1.1 为什么要学习 Ada 语言

目前全世界使用的计算机语言数以百计,一般软件人员都熟悉其中一种或几种,而且用起来得心应手。那么,为什么还要学习 Ada 语言呢?学习 Ada 语言的必要性何在?

概括地说,有以下几个方面的理由:

第一,Ada 是现有的语言中无与伦比的一种大型通用程序设计语言,它是现代计算机语言的成功代表,集中反映了程序语言研究的成果。Ada 的出现,标志着软件工程成功地进入了国家和国际的规模。在一定意义上说,Ada 还冲破了“冯·诺依曼思维模式”(Von Newman Mind-set)的桎梏,它连同 Ada 支持环境(APSE)一起,形成了新一派的所谓“Ada 文化”。它是迄今为止最复杂、最完备的软件工具。

60 年代以后,美国开始出现“软件危机”,主要表现为:软件的响应速度往往不能满足用户要求;经常出故障,可靠性低;对软件的费用无法预测,出现超支;维护复杂、费用较高,并且易于出错;软件常常延期交货,而且比原来商定的功能少;移植性差,即使两个系统需要类似的功能,一个系统的软件也很难用于另一个系统;软件开发工作不能最有效地利用有关的资源,包括处理时间和内存空间,因而效率不高。

出现软件危机的原因是多方面的,但是,程序设计语言的陈旧和程序设计实践的落后,不能说不是一个重要的因素,而且这两者是互相关联的。

和其他很多事物一样,不少人都有一种“先入为主”的倾向,对于程序设计语言也是如此。美国 M. V. Devlin 指出:“没有一个经理愿意要第一代真空管计算机,也很少有人愿意放弃例如 FORTRAN 那样的第一代程序设计语言”。没有高水平的程序设计语言,很难设计出高质量的软件,因此,程序设计语言的陈旧是软件危机的一个重要原因。

Ada 语言是为克服软件危机、软件费用的急剧增长而在国际范围内组织设计的一种新的语言。尽管不能说 Ada 语言可彻底克服软件危机(事实上很难找到这样的语言),但从目前来看,在解决实际问题时,特别是在设计和实现大型实时系统时,Ada 通常是最好

的选择。

第二,Ada 支持现代方法学。60 年代末,美国计算机科学界的著名学者 E. W. Dijkstra 提出了结构程序设计(structured programming)。这种方法可以使程序具有一种合理的结构,以使程序易于理解和维护,便于保证和验证程序的正确性。60 年代末到 70 年代,人们在开发软件的过程中,提出并逐步实现了一系列结构程序设计的技术、方法,例如:只用三种基本控制结构,由顶向下设计的逐步求精,模块化等。Ada 支持结构化程序设计方法。应当说,结构化程序设计在提高程序设计的质量,甚至克服软件危机方面是有一定贡献的,然而随着计算机应用的日益推广,程序规模日趋增大,软件的逻辑结构日趋复杂,结构化程序设计渐渐暴露了自己的不足之处。

大家知道,一个计算机程序基本上可以分为两部分,即数据和操作(核心是赋值语句)。结构化程序设计是一种基于分解的程序设计方法。如果以操作为前提进行分解,则会出现全局数据与局部数据互相使用的情况,即数据耦合;如果以数据为前提进行分解,则存在操作耦合问题。传统的自顶向下的设计技术就其性质来说是强制性的,即它强迫人们把注意力集中在解空间的操作上,很少关心数据结构的设计。因此结构化程序设计比较适宜于顺序方法思考问题,不提供任何标识问题空间固有并发性的工具,而问题空间中的算法通常包含有平行性的计算操作。

为了克服结构化程序设计的缺陷,80 年代开始出现了面向对象的程序设计。这是一种现代的程序设计方法学。它强调程序设计不是一个纯功能的设计技术,认为问题对象起着重要作用,每个对象都有自己的操作集合,它把数据和操作分别封闭在各个对象中,只要控制好各对象之间的通信就可以完成各种操作。也就是说,整个系统(主程序)控制的是对象,而每个对象中的操作和数据是独立的。

Ada 不是面向对象的程序设计语言,但它支持面向对象的程序设计。一般来说,支持结构化程序设计的语言较多,甚至用非结构化程序设计语言也可以进行结构化程序设计,然而支持面向对象的程序设计语言却为数极少。在 Ada 中,可通过程序包来实现面向对象的程序设计。如图 1.1 所示。

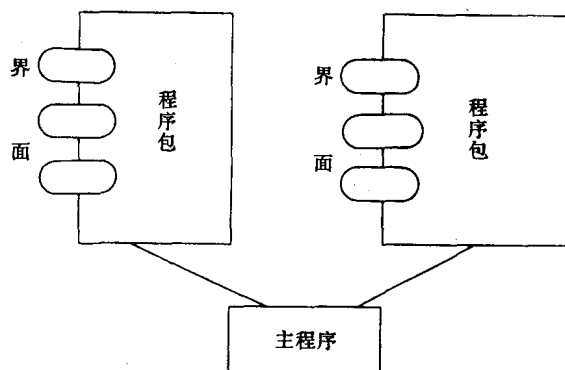


图 1.1 程序包与面向对象的程序设计

在 Ada 中,对象被封闭在程序包中,一个程序包含有各种“资源”,由主程序控制和调用各个程序包的资源,每个程序包都有与外界通信的界面。程序包内的操作是封闭的,与

外界的操作无关。这样就可以实现面向对象的程序设计。

第三,Ada 语言具有结构清晰,修改方便,模块化能力强等特点,所编写的程序可重复使用,而且在减少程序的研制费用和增加程序的移植能力方面,它比任何一种语言都优越。此外,Ada 在数据抽象、模块结构、并发控制、异常处理等方面提出了一整套新概念新方法,这将对今后计算机语言的发展产生重大影响。同时,Ada 语言的应用范围很广,不仅适用于军事实时控制,也可用于数值计算、系统程序设计,以及人工智能和其他方面;它不仅能用于大、中、小型机,而且可用于微型机。专家们预测,Ada 将是今后一个时期内占统治地位的程序设计语言。

Ada 语言满足了其他语言没有满足的要求,具有其他语言不具备的功能,例如任务、类属单元、异常处理以及分别编译等等,这些都是其他语言所望尘莫及的。Ada 语言所具备的能力打破了其他语言强加于程序员的顺序思考问题的方法,从而能适合更广泛的问题空间,为面向对象的设计方法提供理想的语言模型。

第四,Ada 有着十分广阔的前景。一个语言的发展前景如何,是由多方面的因素决定的,美国的 Wegner 提出了一个综合评价语言成就的公式:

$$\text{成就} = \text{设计} + \text{实现} + \text{权势} + \text{需要}$$

用这个公式来评价 Ada,可以说,它在各方面都是十分“成功”的语言。如同前面所讲到的,Ada 是一个设计良好的语言,它体现并实施了许多现代软件开发的方法学,而且经过较长时间反复的规划设计及测评,使得 Ada 设计中潜在错误最少,编译技术已经远远超出早期 FORTRAN 或 COBOL 语言的能力。

Ada 语言得到了具有强大权势的、世界上软件产业的最大用户(美国国防部)的全力支持。美国国防部耗费巨资,组织研制 Ada 语言,目的在于克服软件危机,降低软件费用,代替美国国防部系统中使用的数百种语言,因而竭尽全力推广 Ada,甚至强行规定,所有军用软件一律用 Ada 作为开发工具,否则不予承认。

从全世界来看,计算机软件技术一直处于不断发展中,70 年代占统治地位的是结构化程序设计,而 80 年代则是软件工程。有人估计,软件工程等于给美国增加了 4000 万个劳动力。Ada 语言与早期开发的 FORTRAN、COBOL 等语言不同,它不仅体现了软件工程的原理,而且强化了这些原理,用 Ada 语言编写程序使得软件系统具有清晰、易理解、高效率 and 易维护等特点。Ada 直接提供了支持软件工程各种原理的特征,可以帮助人们开发可修改的、有效的、可靠的和可理解的软件。

1.2 Ada 的特点

Ada 是针对大型、实时、嵌入式系统设计的,它是一种适用于嵌入式系统的通用语言。但在其他领域,诸如系统设计和分布式应用中也很适用。它既适于军用,又适于民用;既适于嵌入式系统,又适于非嵌入式系统。Ada 虽然以 PASCAL 为“母语”,但风格又与 PASCAL 绝然不同,它克服了 PASCAL 的许多局限性和不足。如前所述,Ada 语言体现了许多当代程序设计语言中出现的新思想,反映了软件工程支持系统方面的需要。

尽管 Ada 的“母语”是 PASCAL,但是绝不能把 Ada 看成是扩展了的 PASCAL。总的来看,它有以下一些特点:

1. 程序结构高度模块化

模块化可以使程序结构清晰,易于阅读,便于程序的维护。其他语言的模块化结构通常通过子程序来实现,而 Ada 语言的模块化结构则是通过程序包来实现。程序包是 Ada 的关键特征,它把程序的可见部分(用户界面)和程序包体(具体实现)非常明显地分开,程序员可以规定程序包中的哪些信息是程序包的用户能够使用的,哪些信息对用户是“保密”的。

程序包是有关的类型、常数、子程序及其他资源的集合,Ada 允许将程序包编译好存放到子程序库中。如果用户程序需要用到其中的资源,则可通过名字来访问(即调用)。对程序包内容的修改可以独立进行,对用户程序没有任何影响。

在软件工程中,实行层次式管理。Ada 的程序包为软件开发的层次式管理提供了一种管理工具。任务可以用程序包规格说明(即提供给用户的界面)来下达,而由执行任务的人编写程序包体(即实现程序包的功能)。

Ada 的程序包可作为模块化程序的“构件”,这种“构件”的功能可以越来越强,但保持一个不变的性质,即同外部世界的联系简单,使用也简单。编程的人只要知道某一“构件”的规格说明,就能决定是否选用它,而不必了解内部细节。

程序包可以分别编译,而且程序包的规格说明部分和程序包体部分也可以分别编译,这就为模块化程序设计提供了极大的方便。

2. 分别编译

Ada 由若干程序单元组成,可以分为一系列的编译单元,包括子程序、程序包、子单元等,这些编译单元可以独立地进行编译。如上所述,程序包的规格说明和包体也可以分别编译。分别编译特别适合用来编写大型程序,它有很多优点,例如:

(1)分工方便。大规模的程序往往由许多人分工协作来编写,分别编译就能使程序的编写分块进行,互不干扰。分别编译通过关键字 `separate` 来实现。有了这一功能,就可以先设计好程序的整个框架,然后分别由不同的人编写实现不同操作的程序单元。

(2)便于维护。由于各个模块可以分别编译,因而查错十分方便,查到错误并进行修改后,仅需对修改了的模块进行重新编译,与其他模块互不干扰。

(3)链接方便。可以根据需要对不同的编译单元进行链接。

3. 强数据类型

Ada 是强类型语言,要求所有的对象必须具有唯一的数据类型,而且要在使用前定义,其类型在编译时必须确定,但可以在不同类型之间转换。Ada 的强类型原则迫使程序员在设计程序时必须十分仔细认真,这样不但能更有效地利用存储空间,而且使程序便于调试,只要类型不对,在编译时就要出错。和 PASCAL、PL/1、C 语言一样,Ada 也提供了聚集数据类型。

4. 提供类属设施

在实际的程序设计中,常常需要将同样的逻辑功能应用于不同的数据类型上。在其他语言中,一般要对不同的数据类型编写不同的子程序。作为强类型化的重要补充手段,Ada 提供了一种重要的编译时参数化设施,即所谓“类属设施”。利用这种设施,可以根据逻辑关系写好程序,在使用时将具体数据类型代入(叫做类属设置),这样能缩短程序,提高程序的易读性和功效性。

5. 具有并行处理能力

Ada 最独特和最有吸引力的特征是具有多重任务化设施,支持多个任务的并行处理。多个任务可以在单机上运行,也可以在多机上运行;能在单处理机或多处理机上使用;任务之间还可以通信,进行同步或异步控制。也就是说,利用 Ada 语言编写一个程序,能在同一时间内做几件事。Ada 的任务是真正同时运行的独立进程,它们之间既可同步工作,也可异步工作;它们可以互不相干,也可以互相制约,互相对话及通信。多任务化使得 Ada 可以在单处理机上运行带有多个中断驱动进程的程序。Ada 是第一个提供了实时(中断驱动)逻辑、并发进程和多处理机的程序设计语言。

6. 具有异常处理能力

由于 Ada 是针对嵌入式系统设计的,因而具有异常情况的处理能力,以保证系统的连续运行。异常处理用来处理那些不希望发生,但有可能发生的操作及计算错误等,使得系统不致出事故。当出现错误时自动“引发异常”,并执行处理这些情况的例行程序。Ada 的异常处理功能可以提高程序的可靠性。

7. 实时处理能力

实时性要求程序设计语言具有最快的响应速度,它包括三个方面:有水平较高的编译程序;有中断响应和并行处理能力;Ada 除具有上面所讲的并行处理功能外,还具有机器级存取功能,它提供了将数据地址分配到位级的设施,允许测试与机器有关的控制标志,状态字和外围接口,这对于实时处理是非常有意义的,因为实时处理中的中断处理,一般总是要访问有关的特定设备。

8. 名字可以重载

所谓重载,就是用同一个名字来表示不同的运算符、枚举类型的元素、变量名或子程序等。Ada 允许名字或符号的重载,可以根据上下文将它们区分开,这样就可以用同一个名字来表示几种类似的运算或操作。

9. 可读性强

阅读一个程序的次数往往比写一个程序的次数要多,因此可读性是很重要的。与其他语言相比,Ada 的可读性要强得多。因为 Ada 的程序单元通常都分为说明和实现两部分,这样,不必了解后面的内容,就能读懂程序,而其他语言则很难做到这一点。可读性对于程序的维护具有重要的作用。

10. 易于移植

Ada 的数值数据类型不受机器的限制,可以根据需要用不同数量的字节来表示一个数,这使得一个具体数的表示在各种机器上都是一样的,因而程序更易于移植。

1.3 国外对 Ada 的开发、应用情况

1975 年 1 月,美国国防部开始研制 Ada 语言,1980 年 7 月正式发表了 Ada 语言的标准文本,并将其定为美国军用标准(MIL-STD-1815A)。此后,Ada 语言文本几经修订,于 1983 年被定为美国国家标准(ANSI 1815A)。1987 年 3 月,国际标准化组织接受 Ada 语言为国际标准语言。

由于 Ada 语言并不仅仅是又一种新的计算机语言,而是一整套软件工程方法学的体

现,因此,与其他语言相比,推广 Ada 语言具有特别重要的意义,但同时也会遇到相当大的困难。美国国防部和欧、日各国都把培训和推广作为一项很重要的工作来抓。

从 Ada 语言产生之日起,就一直对 Ada 语言的性能存在着争论。但是,美国国防部坚决支持 Ada 语言。1980 年 7 月,美国陆军曾规定,从 1983 年起,新研制的和要作重大改进的软件都要使用 Ada 语言。美国国防部也曾明确规定,到 1984 年 1 月 1 日进入高级研制阶段的嵌入式计算机和到 1984 年 7 月 1 日进入全尺寸研制阶段的计算机,其软件必须采用 Ada 语言编写。1987 年 3 月 31 日美国国防部命令:新武器系统全部软件的开发都必须采用 Ada 语言作为单一的公用高级语言。1987 年 4 月,美国国防部副部长威廉·塔福特提出要进一步扩大这个命令应用的范围,规定在所有的国防部计算机资源上必须使用 Ada 语言。因此,在 Ada 语言标准化后的短短几年里,它得到了迅速发展。到 1985 年,美国就有 37 个不同的国防计划使用了 Ada 语言。1987 年,空军有 84 项、陆军有 37 项、海军有 24 项计划已使用 Ada 语言开发软件。其中包括空军的先进战斗机(ATA)计划;陆军的全球军事指挥控制系统的信息系统、先进的地炮战术数据系统计划;海军的 F-4J 武器系统训练模拟器、先进战术飞机等计划。

到 80 年代末期,美国大约有 20 多家公司从事 Ada 语言的编译程序工作。到 1991 年,全世界已推出近 300 个经过严格测试和确认的 Ada 编译程序产品,已覆盖各种主流机型。现在的主要工作是研制 Ada 语言的程序设计环境,并通过使用 Ada 语言来解决同软件生产率、性能、正确性证明和移植性有关的问题;利用 Ada 语言研制各种软件生产工具,提高软件生产率、可靠性和自适应能力。

自 1987 年以来,美国及西欧各国推广使用 Ada 已由过去主要靠行政命令而转向主要靠优异的使用效果来吸引用户。到 1987 年底,军方和工业界已积累起 290 多个可重复使用的 Ada 软件包,从而大大提高了编程效率。有 40 多项国防计划使用了 Ada 语言,并取得明显的效果,Ada 语言在国防系统中的市场已超过 18 亿美元。目前 IBM 公司、CRAY 公司都转向采用 Ada 语言,欧洲的 ESPRIT(欧洲研究开发战略计划)已把 Ada 作为各种系统的统一目标语言。

Ada 语言除了军用以外,也正广为民用,尤其是用于航空航天工业。美国宇航局正在使用 Ada 语言为空间站研制软件支持环境;联邦航空局为提高空中交通管制系统的性能,也正在利用 Ada 语言研制高级自动程序。世界上著名的 IBM 计算机公司已开始转向采用 Ada 语言。

除美国外,英国国防部、前西德国防部、欧洲共同体和日本通产省也在合作开发 Ada 语言。在 1984 年 10 月的华盛顿会议上,最高层的北约委员会阐述了自动数据处理政策,指挥和控制数据处理委员会还建议:(1)北约应该选择 Ada 语言,作为指挥控制信息系统的高级语言;(2)北约应继续监视世界 Ada 语言的发展动向;(3)北约应立即成立专门的 Ada 语言办公室;(4)为将来计划的需要,应该培训使用 Ada 语言的人员。1985 年 1 月 25 日,北约国防计划委员会通过了上述建议。为了把 Ada 语言用于北约的数据处理系统并增加 Ada 语言程序设计支持能力,加拿大、丹麦、荷兰、法国、英国、意大利、挪威、西班牙、美国和前西德等 10 个国家成立了专门工作组,并签订了合作协议备忘录。北约组织正式决定采用 Ada 语言作为标准语言。可以预见,Ada 语言作为 80 年代和 90 年代的程序设计语言,将对全世界的计算机软件产业产生重大的影响。

1.4 Ada 的构成

一个 Ada 程序(或 Ada 系统)由一个或多个程序单元组成,每个程序单元可以分别编译,Ada 的程序单元有四种,即子程序、程序包、类属单元和任务。其中子程序分为过程和函数两种,它一般完成独立的操作。程序包是计算资源的集合,它把类型、数据对象、子程序、任务、甚至其他程序包封装起来。类属程序单元为定义对象或操作而建立参数化的模板。任务定义了逻辑上与其他任务并发执行的动作,任务可以在单处理机、多处理机或计算机网络上实现。Ada 各种程序单元的特性和用途如表 1-1 所列。

表 1-1 Ada 程序单元

程序单元	特性	用 途
子程序	顺序的动作	主程序单元 定义功能性的控制 定义类型操作
程序包	资源的集合	声明的命名集合 相关程序单元组 抽象数据类型
任务	并发的动作	并发动作 信息路径 控制资源 中断
类属单元	公共操作	使对象或操作参数化

所有的程序单元具有相似的结构,都由两部分组成,即规格说明(specification)和体(body)。其中规格说明定义了该程序单元用户的可见信息,即界面;而体则包含了程序单元实现的细节,即具体的执行过程。

规格说明部分和体部分可以分别编译。因此,在设计程序时,可以先写程序单元的规格说明,建立该类型的结构,在以后的开发过程中,加上单元的体,并单独细化,以完善整个系统的设计。

在以后的章节中,我们将详细地讨论这些程序单元。

习 题

1-1 与其他流行的程序设计语言相比,Ada 语言有什么特点?

1-2 学习 Ada 语言的必要性何在?

第二章 Ada 词法元素

和其他语言程序一样,Ada 程序由各种词法元素组成,词法元素由一个或多个字符组成。在用 Ada 编写程序之前,必须先了解这些词法元素的规则和含义。

Ada 中的词法元素分为六类,即标识符(包括保留字)、定界符、数值字面量、字符字面量、字符串和注释。数值字面量之间和标识符之间必须用空格(至少一个)或换行符隔开。

本章将介绍 Ada 的各种词法元素。

2.1 字符集

Ada 所使用的字符集与其他语言大致相同。共有 95 个,其中基本字符 56 个,扩充字符 39 个。

基本字符包括大写字母、数字和特殊字符,即

大写英文字母 26 个: A...Z

数字 10 个: 0~9

特殊字符 19 个:

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

空格

除以上基本字符外,Ada 还允许使用 39 个扩充字符,包括:

小写英文字母 26 个: a...z

特殊字符 13 个:

! \$ % & ' () * , + - . / : ; < = > _ | &

字符的顺序规定为:

0 < 1... < 9 < A < B <... Z < a < b <... < z

2.2 标识符

Ada 中允许用户访问的对象的名字叫标识符。它是用来表示程序包、过程、函数、类型、常量及变量等名称的符号。

标识符的构成规则:

- (1) 由字母、数字、下横线构成;
- (2) 字母开头,字母数字序列,字母或数字结尾;
- (3) 不能用下横线结尾;
- (4) 单个字母也可以构成标识符。