

1
计算机科学与
技术丛书

郭浩志 编著

ADA程序设计基础

科学技术文献出版社重庆分社

ADA

程序设计基础

郭浩志 编

科学技术文献出版社重庆分社

68225

89.11 21

ADA程序设计基础

编 著：郭浩志

责任编辑：林云梯

科学技术文献出版社重庆分社出版

重庆市市中区胜利路132号

新华书店重庆发行所发行

中国科学技术情报研究所重庆分所印刷厂印刷

开本：787×1092毫米1/32 印张：14 字数：30万

1989年2月第1版

1989年2月第1次印刷

科技新书目：185—293

印数：1—3500

ISBN7-5023-0321-9/TP·23

定价：4.30元

007525

KL02/05

内容简介

本书以ADA语言(1983年版本)为背景,重点介绍大型软件的基本部件——程序单元的程序设计,同时力求阐明七十年代以来程序设计方法学与软件工程的一些新概念、新思想和新方法在ADA的具体体现。全书突出四种程序单元(子程序、程序包、类属单元和任务)的结构、特点、设计和使用,各程序单元之间以及程序单元与外界之间的各种接口,并结合实例介绍ADA直接支持开发大型软件的自顶向下方法和自底向上方法。每章都配以形式多样的习题。

本书可作为高等院校计算机及有关专业高年级大学生或研究生的教材,也可用作教师及各种应用领域的科研、设计人员的参考书。

计算机科学与技术丛书编委会成员

名誉主任	邹海明	华中工学院
主 任	徐洁磐	南京大学
副主任	王政本	北京大学
	左孝凌	上海交通大学
委 员	(以姓氏笔划为序)	
	庄心谷	西北电讯工程学院
	朱宗正	华东工学院
	李 勇	国防科技大学
	李盘林	大连工学院
	陈 禹	中国人民大学
	陈增武	浙江大学
	张太行	华中工学院
	杨文龙	北京航空学院
	杨祥金	南京工学院
	郑人杰	清华大学
	周冠雄	华中工学院
	胡铭曾	哈尔滨工业大学
	侯广坤	中山大学
	洪声贵	辽宁大学
	袁开榜	重庆大学
	徐君毅	复旦大学
	董继润	山东大学
秘 书	朱树春	南京大学

序 言

计算机科学与技术丛书是由计算机教育学会主编的，它以推广和普及计算机应用、培训计算机人才为其主要目标。

丛书以三个面向为宗旨。**面向基础**，为大专院校提供教材，为工程技术人员提供更新知识、扩大知识面的参考书；**面向应用**，为各行业从事计算机应用的工程技术人员提供实用的设计和编程范例，以资借鉴；**面向提高**，介绍计算机技术发展的新动向，以便及时了解其国内外最新技术。

丛书编委会认为，本丛书力求做到：理论联系实际，既有一定的基础理论知识，又有应用理论解决实际问题的方法和实例；普及与提高相结合，有一定的先进技术，又着眼于为当前应用服务，以满足各层次人员学习和运用计算机的需要。

本丛书力求做到内容新颖，重点突出，科学性强，条理清楚，叙述严谨，简单易懂，以适应自学的要求。

参加本丛书编写的作者，都是在计算机教学、科研、开发与应用第一线工作的同志。既有一定的理论基础，又有丰富的实践经验。相信本丛书会得到广大读者的欢迎。我们真诚地希望广大读者对丛书提出批评和监督，以利我们改进工作，更好地为读者服务。

计算机科学与技术丛书
编委会

前 言

本书是按计算机科学与技术丛书编委会审定的编写大纲编写和审订的。由国防科技大学郭浩志副教授编写，李勇教授任责任编委，齐治昌副教授主审。

ADA的诞生标志着程序设计语言已趋成熟。它经历了两个阶段：第一阶段（1974—1980年）制订语言文本；第二阶段（1981至今）研究编译和支持环境。1983年以来全世界已有近百个编译系统陆续投入使用。ADA总结了多数流行语言的重要概念和方法，吸收了七十年代以来计算机科学和软件工程若干新思想。它的功能完善，讲究效率，语言设计中处处强调可靠性、可维性与可移植性。它最初是为军事部门的嵌入式系统开发的，如今已越来越多地用于开发工业、商业、科学研究和大学教学领域中的大型实时和通用系统。ADA，与其简单地说是新语言，还不如说为软件系统的集成、开发和工程化提供了一个强有力的工具。本书力图使国内读者，尤指高校大学生和研究生能较快地全面熟悉和应用ADA，并在程序设计的能力、水平和素养诸方面得到明显的提高。

编者在总结近几年科研、教学实践经验和体会的基础上编写了本书。编写过程参考了大量国外最新图书资料（见书末所列的参考文献）。全书以标准ADA语言（ANSI/MIL-STD-1815A-1983）为背景，比较系统、深入、准确地讲述了该语言的语法、语义和语用。全书正文部分共分八章。第

一章：ADA程序结构，按模块观点剖析了ADA与早期语言的主要区别；第二章：程序单元的组成，介绍了ADA模块的基本数据类型和流程控制结构；第三章：基本程序单元——子程序；第四章：公用实体的程序单元——程序包；第五章：模板程序单元——类属单元；第六章：并行处理的程序单元——任务；第七章：程序单元的其它外部接口，进一步说明了程序单元与外部联系的其它各种方法，其中包括三种级别的输入输出，程序设计环境，物理层数据的描述，中断，与其它语言的联用等等；第八章：大型软件的开发方法及其应用，重点研究ADA对开发软件系统所提供的方法上的直接支持，并以实例介绍了两种典型方法（自顶向下与自底向上）。第一章简述本书出发点，第二章是语言基础，第三至六章是本书重点，第七章作了必要的扩充，第八章点明主题。

全书重点突出，对四种典型程序单元，均从软件工程和程序设计方法学的角度作了评述。概念引进自然，层次组织合理。注意与传统语言对照。举例典型，习题丰富多样。文字简洁流畅，适于自学。

在本书编写过程中，曾得到计算机科学与技术丛书编委会、科学技术文献出版社重庆分社和国防科技大学领导、教师和学生的大力支持和帮助，在此深表谢意。

由于本人水平所限，编写时间又较仓促，书中缺点错误难免，恳切希望读者多加指正。

编 者

目 录

第一章	ADA程序结构	(1)
1.1	早期语言的模块.....	(1)
1.2	ADA的模块及其基本特点.....	(3)
1.3	ADA程序一瞥.....	(5)
1.4	基本程序结构.....	(7)
	习题.....	(9)
第二章	程序单元的组成	(11)
2.1	基本词法特征.....	(11)
2.2	纯量类型.....	(18)
2.3	顺序控制结构.....	(33)
2.4	选择控制结构.....	(39)
2.5	重复控制结构.....	(45)
2.6	数组类型.....	(54)
2.7	简单记录类型.....	(74)
	习题.....	(78)
第三章	基本程序单元——子程序	(83)
3.1	函数.....	(83)
3.2	过程.....	(93)
3.3	异常及其处理.....	(101)
3.4	子程序说明.....	(106)
3.5	重载.....	(108)
3.6	子程序的定义域、作用域与可见性.....	(111)

3.7	子程序的转移	(112)
3.8	子程序的副作用	(113)
3.9	嵌入式子程序	(114)
3.10	对ADA子程序的评述	(114)
	习题	(116)
第四章	公用实体的程序单元——程序包	(118)
4.1	程序包机制	(118)
4.2	异常说明与异常引发	(125)
4.3	私有类型与延迟常量	(133)
4.4	判别记录类型	(143)
4.5	存取类型	(156)
4.6	派生类型	(175)
4.7	再谈数值类型	(183)
4.8	程序包的定义域、作用域与可见性规则	(198)
4.9	再命名规则	(202)
4.10	对程序包的评述	(202)
	习题	(203)
第五章	模板程序单元——类属单元	(207)
5.1	类属单元的说明、实例化与使用	(208)
5.2	类属类型参数	(213)
5.3	类属对象参数	(219)
5.4	类属子程序参数	(221)
5.5	嵌套类属单元	(225)
5.6	无参类属单元	(226)
5.7	类属与异常	(230)
5.8	对类属单元的评述	(231)
	习题	(232)

第六章 并行处理的程序单元——任务	(236)
6.1 并行问题的提出	(236)
6.2 任务的定义	(237)
6.3 任务的激活、执行与结束	(239)
6.4 任务间的同步与通讯	(241)
6.5 任务的延迟执行	(248)
6.6 对接收语句的控制	(251)
6.7 对任务入口调用的控制	(264)
6.8 任务类型与任务对象的建立	(268)
6.9 任务的终止、夭折与异常	(275)
6.10 任务的优先执行	(278)
6.11 对ADA任务的评述	(287)
习题	(289)
第七章 程序单元的其它外部接口	(293)
7.1 输入输出概述	(293)
7.2 二进制输入输出	(295)
7.3 正文输入输出	(301)
7.4 物理层数据表示法规范说明	(326)
7.5 中断	(325)
7.6 低级输入输出	(337)
7.7 代码语句	(338)
7.8 SYSTEM程序包	(348)
7.9 与其它语言的接口	(342)
7.10 无检查的程序设计	(343)
7.11 ADA程序设计支持环境概述	(345)
习题	(348)
第八章 大型软件的开发方法及其应用	(351)

8.1	程序库.....	(351)
8.2	分别编译.....	(353)
8.3	自底向上的开发方法及其应用.....	(355)
8.4	自顶向下的开发方法及其应用.....	(368)
	习题.....	(377)
附录一	ASCII 字符集.....	(380)
附录二	保留字.....	(385)
附录三	预定义标识符.....	(386)
附录四	预定义属性.....	(388)
附录五	预定义编译注解.....	(395)
附录六	预定义环境.....	(397)
附录七	语法规则.....	(420)
附录八	小词典.....	(429)
	主要参考资料.....	(437)

第一章 ADA程序结构

为解决七十年代以来的软件危机，人们一直在探讨如何利用人类科学知识和技术原理来设计和构造软件。可以预料，计算机软件以一个独立工业部门的崭新姿态出现在工业界，为时也许不会太远了。这一发展趋势表明，程序不再是一种技艺。人们应该用一种科学的、工程的观点去认识、应用和发展软件这门学科。

大规模集成电路在硬件系统设计、维护与生产中已获得广泛的应用。它不仅对建造大型系统提供了各种基础构件，而且对提高计算机系统的可靠性、生产效率和性能价格比起着无可争辩的作用。这种模块化思想是否也可以应用于开发大型软件系统呢？

在面向过程的通用语言中，ADA无疑是迄今为止最完善、最强有力的程序设计语言。ADA语言文本的诞生，APSE环境的开发都无愧为应用软件工程原理和方式开发系统的范例。ADA从语言一级上对软件（尤其是大型软件）的开发以及对提高软件的可靠性、可维性和可移植性都提供了直接而有力的支持。因此，在解决各种实际问题时，如果还完全照搬当今流行语言的方法使用ADA，那当然大大降低ADA的意义。任一软件系统均可被看作是一些逻辑相关的程序模块的有机结合。ADA语言的模块化结构，以其现代程序概念和思想，比较成功地解决了软件的集成、开发和研制问题。

1.1 早期语言的模块

在程序语言发展早期，人们就已认识到程序模块的优点，如程序结构清晰、紧凑、易读、易编、易分工、节省内存，等等。模块方法为语言设计者和用户所一致推崇。当今最流行的语言，如COBOL, FORTRAN, BASIC, ALGOL 60和PASCAL等都具有模块机制。尽管它们的模块名称不一，如子程序、过程、函数、分程序、节、段，但都具备一些共同的基本特征，如功能明确单一，模块调用和信息传递方法规范，主张单入口单出口，易于构造更大的程序等。然而，七十年代中期以来，许多学者在世界范围内有针对性地作了比较全面、认真和科学的调查与分析，发现早期语言的缺点还很多，远远不能适应当今软件发展的需要。我们仅从软件开发的角度的，指出模块三个方面的问题：

1. 模块本身的可靠性问题

对于一个模块来说，用户应该知道的和不该知道的事情往往不太明确，特别表现在模块对外的接口与内部实现细节在结构上多少有点混淆。如当FORTRAN的子程序中使用了公用区时，不同调用的效果可能不同，甚至可能产生名字冲突，直接影响了程序的正确性。又如ALGOL60的过程由于允许出现全程量，模块自身的正确性依赖于其上下文，不仅在使用之前无法确认，而且在编译时也无法进行一致性检查。这个问题，一方面说明对应给予隐蔽的信息没有安全可靠的保护措施，另一方面也反映了用户对模块的使用规则不够严密。

2. 模块间的关系问题

早期语言程序模块之间的关系并不平等，从而导致使用上缺乏灵活性。例如，PASCAL语言的主程序与各子程序（无论是并列的或嵌套的）之间的相互访问受到各种限制。这大大影响了模块的可用性与可构造性。

3. 构造软件系统问题

早期语言在构造大型软件方面是很不适应的。例如，FORTRAN语言的模块结构最多只能是二级的，即只能将主程序同函数辅程序或子程序辅程序联结在一起。ALGOL60语言的分程序结构，对可见性的控制不太精确。PASCAL语言虽是第一个结构程序设计的语言，但它只适应自顶向下逐步求精方法，而且对用户来说，这种方法只能说是提倡性的，语言本身并未直接提供相应的机制，程序员完全可以不采用该方法也能编写软件程序。

1.2 ADA的模块及其基本特点

用ADA开发的软件，可以由一个或多个模块组成。这种模块是ADA程序的基本部件，我们称之为程序单元。ADA共有四种程序单元。除了已为大多数语言所必备的子程序（ADA包括过程与函数）外，还有程序包、任务和类属单元。四种程序单元各有不同的特点和用处。子程序定义一个具体算法，它给出完成特定功能的一个操作或一序列计算步骤；程序包由逻辑上相关的一组实体（指类型、常量、变量和子程序等）组成；任务定义并行计算；而类属单元则定义带参数的程序单元，本身不能执行，但可从它获得各种

可供使用的特定子程序或程序包。

尽管上述四种程序单元在功能和用处上各有不同的特点，但它们的基本结构却十分相似。每一个程序单元都分成两部分：规范说明部分和体部分。规范说明部分定义供外部模块访问的各种实体（包括该实体的名字、参数和其它信息）。体部分包括为实现该模块功能的具体算法和细节（代码、数据和类型等）。体不允许其它模块访问。

ADA将模块的规范说明与体分离和封装的方式，使用户只可了解、获取允许他访问的东西，而隐蔽了不该让他知道的所有实现细节。因此用户就能精确控制模块的可见性。而且，只要对外接口不变，对体部分的任意修改和重新编译既不影响对应的规范说明，也不波及访问它的其它模块。

编译程序不仅可以分别编译诸程序单元，而且还可分别编译模块的规范说明及其对应的体。更重要的是，分别编译时还可作完整的一致性检查。这样，开发软件时，易于将它分成很多结构上相互独立的部分，由很多人分头设计、编写、调试和维护。此外，ADA提供的库单元、子单元机制，从语言一级上直接支持自底向上和自顶向下两种程序设计或软件开发方法，有助于开发大型软件。

ADA的模块方法不仅继承传统模块的可读性、可理解性等优点，而且在提高可靠性、可维性、可移植性、可构造性、可用性和标准化程度等方面，都向着使软件成为一个崭新工业部门的方向，扎实地迈进了一大步。当然，以后我们将会陆续知道，ADA上述特性不仅体现在模块机制上，还渗透到各种语法成分之中。

1.3 ADA程序一瞥

我们先来编写一个求两整数之和的简单ADA程序ADDITION。它先将两个整数分别读入变量X和Y中，然后在新的一行，以直观形式输出X与Y之和。程序如下：

```
with TEXT_IO; use TEXT_IO;
procedure ADDITION is
  X, Y: INTEGER range 0...2000;
begin
  GET(X);
  GET(Y);
  NEW_LINE;
  PUT("THE SUM OF");
  PUT(X); PUT("+"); PUT(Y);
  PUT("="); PUT(X + Y);
end ADDITION;
```

这是仅由一个过程构成的程序。该过程以保留字procedure打头，以保留字end和分号(;)押尾。procedure之后是过程名ADDITION和保留字is，end与分号之间可以插入同样的过程名。is与end之间为过程体。

过程前的with(保留字)子句，列出了程序包TEXT_IO的名字。表示位于with子句后的过程ADDITION用到了库单元TEXT_IO中的实体。TEXT_IO是一个预定义程序包，它提供各种正文输入输出操作。

is与begin之间是过程体的说明部分，它给出各种实体