

高等学校教材

Ada 语言

杨孝宗 编

哈尔滨工业大学出版社

Ada 语言

杨孝宗 编

哈尔滨工业大学出版社

内 容 提 要

本书系统地讲述了Ada语言的语法规则和Ada程序设计技术与原理。内容主要包括Ada的语句、类型化、程序单元、任务、类属、异常处理以及输入输出等等,同时还提供了若干可以运行的Ada程序实例。为了便于练习及Ada语言的开发,书中还给出了相应的习题和与Ada语言相关的附录材料。

本书是面向研究生和大学本科生的教材,也可供从事计算机科学与工程科技工作者参考。

Ada 语 言

杨孝宗 编

*

哈尔滨工业大学出版社出版
新华书店首都发行所发行
哈尔滨市外文印刷厂印刷

*

开本 787×1092 1/16 印张21.125 字数442 000

1989年12月第1版 1989年12月第1次印刷

印数1-1 000

ISBN 7-5603-0215-7/TP·14 定价4.50元

出版说明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校、中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978年至1985年，已编审、出版了两轮教材，正在陆续供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻“努力提高教材质量，逐步实现教材多样化，增加不同品种、不同层次、不同学术观点、不同风格、不同改革试验的教材”的精神，我部所属的有关七个高等学校教材编审委员会和两个中等专业学校教材编审委员会，在总结前两轮教材工作的基础上，结合教育形势的发展和教学改革的需要，制定了1986~1990年的“七五”（第三轮）的教材编审出版规划。列入规划的教材、实验教材、教学参考书等近400种选题。这批教材的评选推荐和编写工作由各编委会直接组织进行。

这批教材的书稿，是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会（小组）评选择优产生出来的。广大编审者、各编审委员会和有关出版社为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还会有缺点和不足之处，希望使用教材的单位、广大教师和学生积极提出批评建议，共同为不断提高工科电子类专业教材的质量而努力。

机械电子工业部电子类专业教材办公室

前 言

本教材系按机械电子工业部的工科电子类专业教材1986—1990年编审出版规划,由“计算机与自动控制”教材编审委员会“计算机”编审小组征稿、推荐出版,责任编辑委陈火旺教授。

本教材由哈尔滨工业大学杨孝宗编写,北京工业大学丘玉圃教授担任主审。

本课程的参考学时数为40学时,其主要内容为Ada语言的规则以及程序设计方法。本教材从现代化软件工程学的角度,系统、详细地阐述了Ada的设计思想。书中除了介绍Ada与其他语言的相似之处外,较为详尽地讲述了抽象型数据结构、Ada的检测机构,如Ada的典型化、特殊类型、程序包和子程序、异常处理等,重点讲述了Ada支持的并行处理能力,如task,以及支持大型程序设计的若干机制,如类属、分别编译、编用等等。最后,讲述了输入和输出的有关内容。为了便于学习和实践,书中给出了相应的习题和一定数量的Ada程序举例。这些例子是在VAX/VMS以及IBM-PC/XT上,经过调试运行的,有些则是从大型程序中选择一部分,以供参考。本教材的后一部分给出了有关的附录材料,包括Ada的环境、属性、编用等等。使用本教材时,应注意突出重点。本教材适合于有一定软件基础的学生学习,因此,属于Ada与其他语言的相似部分可以简化,重点放在Ada是如何支持现代化软件工程的这一方面。学习中应注意,不能把Ada当作一般的语言来看待,应该突出它的设计思想及其特点。Ada是大型的、较为复杂的语言,学习中应多结合一些实例,而不能仅靠Ada手册,因此,要有一种好的学习方法。

本教材由滕继全同志编写了第十四章及其习题。林彬同志编写了第一章至第八章的习题。邢国光同志编写了第九章至第十五章的习题。林彬、邢国光二同志又编写并调试了“Ada程序举例”中的各条程序。滕继全、张晓梅同志参加了整理工作。本书编写中,得到了陈火旺教授、郭福顺教授、丘玉圃教授以及王振宇高级工程师的热情帮助。在此对各位的合作和帮助表示诚挚的感谢。

由于编者水平有限,书中难免还存在一些缺点和错误,殷切希望广大读者批评指正。

编 者

075124/2

目 录

第一章 Ada 概论

第一节 Ada的背景	(1)
第二节 Ada的开发史	(3)
第三节 Ada的特点	(4)
第四节 Ada程序结构	(5)
习题一	(7)

第二章 词法单元

第一节 基本符号	(9)
第二节 标识符	(9)
第三节 保留字	(10)
第四节 数值字面量的表示	(11)
第五节 字符字面量的表示	(12)
第六节 注释	(12)
习题二	(12)

第三章 数据类型及对象声明

第一节 Ada的类型 (type)	(14)
第二节 名字与表达式	(16)
第三节 对象的声明及赋值	(19)
习题三	(21)

第四章 标量类型

第一节 整数类型	(24)
第二节 实数类型	(26)
第三节 浮点类型	(26)
第四节 定点类型	(28)
第五节 字符类型	(29)
第六节 枚举类型	(29)
第七节 布尔类型	(33)
习题四	(33)

第五章 子类型和派生类型

第一节 约束	(35)
第二节 子类型	(36)
第三节 派生类型	(38)
习题五	(40)

第六章 控制语句

第一节 语句概述.....	(42)
第二节 块语句.....	(43)
第三节 if语句(条件语句).....	(45)
第四节 case语句(情况语句).....	(47)
第五节 loop语句(循环语句).....	(49)
第六节 转移语句.....	(55)
习题六.....	(55)

第七章 子程序

第一节 过程.....	(57)
第二节 子程序的参量.....	(59)
第三节 函数.....	(62)
第四节 重载.....	(64)
第五节 子程序参量的可见性.....	(67)
第六节 子程序的应用.....	(68)
习题七.....	(71)

第八章 程序包

第一节 程序包结构及封装.....	(74)
第二节 私有类型.....	(78)
第三节 分别编译.....	(81)
第四节 程序包的应用.....	(82)
习题八.....	(84)

第九章 组合类型

第一节 数组.....	(86)
第二节 数组类型.....	(90)
第三节 无约束型数组类型.....	(93)
第四节 数组参数及数组的分片.....	(95)
第五节 字符类型和字符串类型.....	(99)
第六节 动态数组和数组操作.....	(100)
第七节 记录类型.....	(102)
第八节 带判别式的记录类型.....	(105)
第九节 变体记录类型.....	(108)
习题九.....	(110)

第十章 访问类型

第一节 访问类型.....	(111)
第二节 分配算符及访问对象的引用.....	(113)
第三节 用于访问类型的连接操作.....	(116)
第四节 访问类型的典型应用.....	(116)

习题十	(122)
第十一章 类属程序单元	
第一节 问题的提出	(124)
第二节 类属程序单元的结构及例示	(126)
第三节 类属程序单元的类型参量及其匹配规则	(132)
第四节 类属程序单元的形式参量	(138)
第五节 类属程序单元的子程序参量	(139)
第六节 类属程序单元的应用	(142)
习题十一	(145)
第十二章 任务	
第一节 Ada任务的规范说明和任务体	(146)
第二节 任务类型及其对象	(149)
第三节 任务入口及其调用	(152)
第四节 接收语句	(154)
第五节 任务间的会合与信息交换	(156)
第六节 延迟语句及多任务的时间安排	(161)
第七节 选择语句	(162)
第八节 任务的终止	(172)
第九节 任务的运行状态	(174)
第十节 Ada任务的应用	(179)
习题十二	(190)
第十三章 异常处理	
第一节 异常的声明和引发	(193)
第二节 异常的处理	(197)
第三节 异常的作用域	(200)
第四节 异常处理的应用	(205)
习题十三	(208)
第十四章 输入与输出	
第一节 Ada输入与输出的基本概念	(210)
第二节 顺序文件与随机文件的输入与输出	(211)
第三节 顺序文件与随机文件的管理系统	(213)
第四节 文本文件的输入与输出	(213)
第五节 IO_EXCEPTION的规范说明	(228)
第六节 低层次输入输出	(228)
习题十四	(228)
第十五章 Ada的大型程序设计	
第一节 大型程序的结构设计	(231)
第二节 大型程序的编译	(233)

第三节 面向对象的Ada程序设计.....	(237)
-----------------------	---------

习题十五.....	(241)
-----------	---------

附录A Ada程序举例

A. 1 求全班某课程的平均成绩.....	(243)
A. 2 枚举类型的属性.....	(243)
A. 3 字符类型的例子.....	(244)
A. 4 打印负数.....	(245)
A. 5 统计全班成绩总分和平均分数.....	(245)
A. 6 打印0~31能被3整除的数.....	(246)
A. 7 计算工业产值.....	(247)
A. 8 逻辑运算举例.....	(248)
A. 9 求质数程序.....	(248)
A.10 统计选票程序.....	(249)
A.11 解二元一次方程.....	(250)
A.12 解一元二次方程.....	(251)
A.13 排序程序.....	(252)
A.14 表头管理程序.....	(254)
A.15 过程形参使用的举例.....	(256)
A.16 发放工资程序.....	(257)
A.17 整数运算和输入输出程序.....	(258)
A.18 任务通信程序(信息交换).....	(260)
A.19 读写顺序文件.....	(261)
A.20 关于颜色的对话.....	(262)
A.21 整数栈管理问题.....	(263)
A.22 典型筛的问题.....	(265)
A.23 编译程序的验证程序.....	(266)
A.24 保护时间程序.....	(268)
A.25 树的处理问题.....	(273)
A.26 拷贝程序问题.....	(275)
A.27 队列处理问题.....	(278)
A.28 Ada与SMG的接口程序.....	(279)
A.29 软件表决器程序.....	(284)
A.30 银行管理模拟程序.....	(289)

附录B 预定义环境

B. 1 程序包STANDARD.....	(296)
B. 2 程序包CALENDAR.....	(300)
B. 3 程序包SYSTEM.....	(301)
B. 4 类属程序包SEQUENTIAL_IO.....	(301)

B. 5	类属程序包DIRECT_IO.....	(303)
B. 6	程序包TEXT_IO.....	(304)
B. 7	程序包IO_EXCEPTIONS.....	(310)
B. 8	程序包LOW_LEVEL_IO.....	(311)
B. 9	类属过程UNCHECKED_DEALLOCATION.....	(311)
B.10	类属函数UNCHECKED_CONVERSION.....	(311)
B.11	类属过程UNCHECKED_DEALLOCATION.....	(311)
附录C	Ada的预定义语言编用.....	(312)
附录D	Ada预定义语言属性.....	(315)
附录E	Ada语法总结.....	(320)
参考文献		(326)

第一章 Ada 概 论

程序设计语言的领域，一直十分活跃。近几年来，各种专用语言不断涌现，致使语言的种类大增。总的看来，这种现象对计算机的发展是不利的。Ada语言的出现，企图结束这种局面。因此，Ada不是一个普通的语言。它的出现以及它的未来，一直被计算机界密切地关注着。

第一节 Ada的背景

Ada的出现是必然的，也是被迫的。说它必然，是从计算机总的发展来看的。面对飞速发展的计算机技术，除了硬件要发生变革之外，软件也必然随着发生变革。计算机系统正向非冯·诺伊曼系统发展，软件则必然也要转向以非冯·诺伊曼思维方法为基础的软件工程上来。Ada的出现，正是适应了这一趋势。

说它的出现是被迫的，主要是因为软件危机的解决急需象Ada那样的新工具。而且，目前在军用领域中占有极其重要地位的嵌入式系统，对Ada的需要也极为迫切，因此，首先由美国国防部提出了Ada的研制计划。

一、软件危机

众所周知，由于计算机的飞速发展，以及应用范围的不断扩大，人类对计算机的依赖性越来越强。因此，对计算机的要求也就越来越高，计算机所要解决的问题也越来越复杂。而现有的软件工具仍然是以数学计算及以人的技术水平为主要基础的传统方法。而人类本身对问题复杂性的理解、处理方法都是有限的。尤其是对于大型系统来说，仅靠人类自己的技巧是无能为力的。

因此，软件所暴露的问题越来越严重，而且形成了一个恶性循环的局面。这也就是所说的软件危机。

软件危机的主要表现为：不能反映用户的需要、价格昂贵、使用不灵活、维护困难，而且不可靠，同时，交货不及时、移植性差，尤其是不适应大型系统的需要。

我们常用的语言，比如FORTRAN、COBOL、C等等，已经不能胜任现代化的要求了。因为它们的使用局限性很大，基本上还停留在简单指令流的状态下。也就是说，传统的软件无法满足复杂的现代化软件工程的要求，而且它还阻碍了计算机技术的发展。

速度问题。这是实时系统要求最迫切的问题。现代化的控制系统，往往要求大幅度地提高速度。可是，在目前，有的实时系统甚至使用象FORTRAN这样的语言。FORTRAN是为科学计算而设计的，因此它无法满足要求。

费用问题。人们普遍认为软件费用过于高昂。经过分析,我们不难理解,当今的软件是靠软件工作者象艺术家那样拿着落后的工具一点一点地雕刻出来的。这要花去大量的人年工作量。而软件本身的费用又是不可预测的,近年来,呈现着不断增加的趋势,这也是人们普遍担心的问题。

可靠性问题。软件危机的另一重要表现是它的可靠性太低。人们往往把可靠性的注意力放在硬件系统上。认为只要硬件可靠,就能保证系统的可靠。软件本身的可靠性问题往往被忽视。由于计算机的大量推广应用,软件的复杂程度越来越高,程序员的水平相对之下也显得越来越低,从而造成了软件的出错率非常高。而软件的检测能力、保护措施、错误处理能力都非常低,因而,往往一出现软件错误,则束手无策。这些问题直接限制了软件的使用。

可维修性问题。目前,软件的可维修性很低,因为程序的可读性太差。如果不是程序设计者本人,而是由别人来维修一个现存的软件系统,首先就要遇到阅读问题,难以做到熟练地掌握一个系统,维修效果很难得到提高。尤其是对于大型软件系统来说,困难很大,往往是一个待维修的软件问题要拖很长时间,花很高的维修费才能维修完毕。

生产及使用问题。软件生产效率太低。由于软件的单件化生产、通用单元的局限性,人们只能靠“堆积”形成自己的系统,使软件不能最好地发挥现有资源的作用。

总之,软件危机主要表现为,传统的软件工具已经无法处理复杂的问题,从而阻碍了计算机的发展。

二、软件危机产生的原因

前面归纳了软件危机的主要表现,下面进一步分析它产生的原因。

从美国的情况来看,软件危机的主要原因有如下几个方面:

①现有语言种类太多

据统计,美国现有近450种计算机语言,这样多的语言必然要占据大量的存贮空间,要花去大量的培训费,而且软件的移植性和兼容性很差,从而造成了美国的软件费用逐年大幅度增长的局面。

②传统的冯·诺伊曼思维方式的局限性

传统的冯·诺伊曼思维方式已经跟不上现代化的需要了,它无法有效地支持、管理解的复杂性。现代化的软件系统大而复杂,急需要有一种突破性的新方法、新工具。

③组织上和学术上的分散性

这种分散性造成了大量的重复工作,尤其在国防系统中,使工作统一十分困难。这同样也造成了软件费用的增加。

④人类本身的局限性

这一局限性使得人类自己在当今越来越复杂的软件系统面前变得软弱无力。比如,对于一个大型系统的设计来说,仅靠一个人无法完成,需要很多人参加。但是,众多人参加的组织,协调问题也是一个难题。于是,新的工具、新的方法学就显得非常迫切了。

总之,仅靠人类自己的技巧,靠传统的思维方式,已经无法支持现代化的软件设计

方法学, 必须突破人类自身的局限性, 尽早想办法摆脱软件危机所造成的被动局面。

三、Ada与嵌入式系统

如上分析, Ada是在软件危机的情况下提出来的。具体地说, 它的提出是为了在美国国防系统中, 找出能够起统一作用的高级语言, 为嵌入式计算机准备一个强有力的语言工具。

所谓嵌入式计算机, 是指“嵌入”在一个大型系统中的计算机系统, 比如防卫系统、大型武器系统、过程控制系统等等, 都是这样的系统。计算机部分是整个系统的控制中心, 是一个关键部分。嵌入式计算机的主要特征应该是:

①实时性强, 响应速度快。

②具有并行处理及分布式处理的能力。这不仅能解决速度问题, 而且还将解决、处理解的复杂性问题。

③高可靠性问题, 计算机本身应该具备足够的容错能力。

Ada所具备的种种特点, 与嵌入式计算机几乎是一一对应的。几年来的实践证明, Ada是嵌入式计算机系统的有效语言。

第二节 Ada的开发史

Ada的发展史大体上可以分为两个阶段: 分析阶段和设计阶段。

一、分析阶段

七十年代初期, 美国国防部就注意到了软件费用不断增高的趋势。1973年, 美国国防部用于计算机的总支出为75亿美元, 其中, 软件费用就占46%, 而嵌入式计算机的软件支出, 又占总软件支出的56%, 超过了19亿美元。费用高的重要原因之一, 就是语言种类过多(已超过450种)。同时又注意到, 大多数软件费用是和嵌入式计算机系统相联系的。于是, 美国国防部把注意力集中到嵌入式系统上来了。

1975年, 美国国防部成立了一个联合机构——高级语言工作组, 计划只用一种高级语言。这时, 美国国防部下令中止了对其他新语言的研究和开发, 以便于该组织的工作。

该工作组根据国防部对语言的需求, 经过反复调查和修改, 提出了一个又一个文件:

①1975年4月公布了“稻草人”(STRAWMAN)文件, 提出定性的需求条件。

②1975年8月公布了“木头人”(WOODENMAN)文件。

③1976年1月公布了“锡人”(TINMAN)文件。有数百人参加了有关的工作。尽管争论很大, 但是某些观点却取得了统一。那就是未来的语言应该是支持现代化软件方法学的。在这一阶段, 对现有语言作了归纳分析, 分为“不可采纳”的、“不太合适”的以及“可作基础”的三类语言。其中, 选择PASCAL、PL/1和ALGOL_68作为未来语言的基础。

④1977年1月, 在“锡人”文件的基础上, 又公布了“铁人”(IRONMAN)文件。对未来的语言需求做了进一步肯定。

在整个分析阶段, 美国国防部决定开发一种通用高级语言, 并且提出了明确的要

求。

二、设计阶段

美国国防部认为，未来的语言应该有能力作为军方标准语言。为了使该语言更有生命力，而且获得更大的经济效益，又希望语言能为民用部门所采纳。因此，决定在国际范围内进行投标。

1977年4月，在国际范围内公布、征集新语言方案。共有17个组织参加投标。几个月之内，选出了四家提出的方案。从1977年8月开始，这四家又继续进行开发。

1978年2月，进行国际性的评审，选中了其中两家的方案。

1979年3月，在第二次公开评审以后，选中了HONEYWELL BULL公布的方案。该方案是以法国人为首的法国分公司设计的。为纪念世界上第一位程序员、数学家 Augusta Ada Byron (Lovelace伯爵夫人，1815~1851)，该新语言取名为Ada。

与此同时，又开展了Ada环境的设计，也提出了一系列的文件。经过多次反复修改，于1980年7月，把Ada手册送交高级语言工作组，完成了新语言的设计工作。

1980年12月，美国国防部撤消了高级语言工作组，成立了Ada工程联合办公室（即AJPO）负责Ada的工作。1981年1月，将Ada作为美国国防部标准语言。1983年2月，Ada作为美国国家标准。

总之，经过近十年的时间，几千位计算机科学家的努力，Ada正式问世了。又经过数年国际范围内的推广应用，证明Ada是成功的。

第三节 Ada的特点

Ada是一个大型的、功能很强的语言。与其他语言相比，其特点是：

一、实时性强

实时性对语言的要求主要体现在：

①编译水平要高。比如最佳的实现路径，最短的代码序列等等。

②中断响应要快。

③具有并行处理的能力。

Ada具备了以上能力。

二、并行处理

Ada的一个程序单元——任务(task)，是一个活动单元，有其独立性。多任务间又有通信的能力。因此，多任务系统很好地支持了并行处理的能力。这一点是Ada所独有的。

三、提供抽象化设计的能力

Ada的基本程序单元都包含有公开部分和保密部分，使某一些实体的使用特征对外可见，但是具体实现的细节，则被掩盖起来。这一能力直接支持了现代化的软件工程设计方法学。

四、提供类属程序单元

程序设计中，经常要碰到相同的逻辑功能应用于不同的数据类型上。一般的作法是对不同的类型写不同的子程序。而Ada的类型程序单元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。这就给编写程序提供了一个可多次使用、面向多种类型的单元，从而极大地提高了软件生产率。

Ada的类属是计算机语言的一个重大变革。

五、强类型化

Ada是强类型化语言。所有对象在使用之前，必须声明它的类型。这给检测系统带来了很大方便。一旦类型用错，编译系统就能查出其错误。

六、提供程序包结构

类型、对象、子程序、任务放在程序包内，主程序只须调用程序包的内容即可。通常，可以把逻辑相关的实体集合到一个程序包里。因此，程序包的使用为模块化程序设计提供了方便。程序包有公开部分和保密部分，在公开部分中，也开辟了保密区，程序员可以据此选取实体的可见性。

Ada的程序包又提供了分别编译的能力。方便了大型程序设计的要求。

七、分别编译的能力

各程序单元可以分别编译。其优点是：

- ①分工方便。大型程序设计可分块进行。
- ②查错方便。
- ③联接方便。

八、异常情况处理的能力

当程序运行出错时，可以提请系统注意。Ada的异常情况处理可以使程序员根据情况对错误进行处理。Ada的这种能力提高了程序的可靠性，并能很好地支持容错设计技术。

九、易读性

阅读一个程序，往往比写一个程序要频繁得多。因此，可读性是很重要的。Ada的标识符、Ada的名字、语句格式、书写，都很接近自然语言，因此，阅读和书写Ada程序都是非常方便的，这很便于程序的维护和移植。

第四节 Ada程序结构

Ada程序是模块化的结构，这种结构方式不仅方便了程序设计，能提高软件生产率，而且很好地支持了大型程序设计。

一、简单的Ada程序举例

下面是一个简单的但却是完整的程序，

```
with TEXT_IO, use TEXT_IO,           --上下文说明
procedure DOUBLE is
    package INT_IO is new INTEGER_IO(INTEGER);
    use INT_IO,
    X, Y, INTEGER;
begin                                  --体的开始
    GET (X);                          --取一个数
    Y := X * 2;                       --乘 2 再赋值
    PUT (Y);                          --打印结果
    NEW_LINE;                         --换行
end DOUBLE;                          --体结束
```

这是一个过程 (procedure)，过程名为DOUBLE，它的功能是读入一个数，将此数乘以 2，再打印出结果来。

一个Ada程序是由许多基本单元组合而成的。现看一下该程序的组成单元。

①保留字

本程序中用小写字母写的都是保留字，例如，with、use、procedure、is、……，编译系统认识这些字的意义。

②定界符

“，”、“;”、“:”、“:=”、“(”、“)”、“*”等。

③系统预定义名

例如，INTEGER、TEXT_IO都是预定义名，编译系统是认得的。

④系统提供的软件单元

例如，TEXT_IO、INTEGER_IO、PUT、GET、NEW_LINE等都是系统提供的单元。

⑤用户定义的名

例如，DOUBLE、X、Y、INT_IO。

⑥常数

例如，常数 2。

⑦注释

用户所写的有关意义的说明，用两个相连续的上横线组成，例如“--”。编译系统遇到“--”将不理睬其后面的内容。注释行的终止是本行的末尾。注释应由英语写成，这里为便于理解而译成了中文。

程序分成四部分：上下文说明部分（第一行）、程序说明部分（第二行）、声明部分（第三、四、五行）、体部分（从begin到end）。

上下文说明通知Ada编译程序，在程序中，要用到系统的哪几个软件包。比如这里

要用到TEXT_IO软件包。

程序的说明部分给出程序名及程序单元种类。这里给出procedure DOUBLE。声明部分则包括所用对象的声明、类型的定义。比如，声明了X和Y，类属程序包例示后的新名字等等。从begin到end之间为具体的执行语句序列。每个语句的末尾写一个“;”。

二、Ada程序结构

Ada程序由一个或多个分别编译的程序单元组成。通常，每一程序单元由两部分组成：说明部分和体部分。

Ada的程序单元有四种：

子程序：过程（procedure）、函数（function）；

程序包（package）；

类属程序单元（generic）；

任务（task）。

程序单元的划分如图1-1所示。

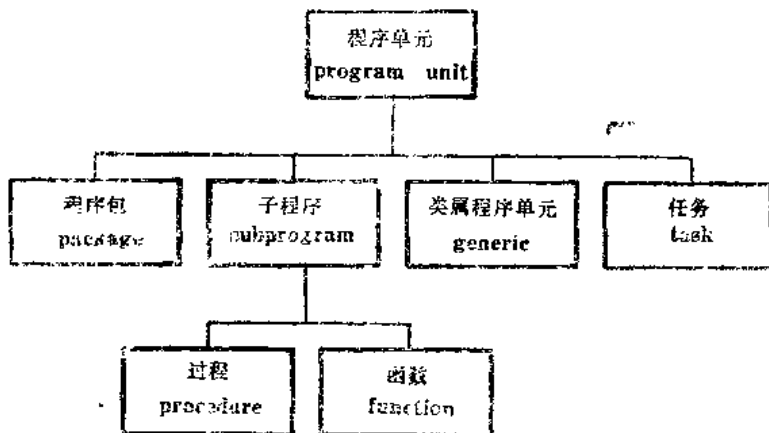


图1-1 程序单元的划分

程序包是逻辑相关的实体结合，子程序可以放在程序包里。子程序包括函数和过程两种，过程是一系列操作的集合，而函数可以返回一个结果。类属程序单元是一个公共模板，代入实际类型（或对象、子程序）之后，可以作为实用的单元来使用。任务是活动单元，是并行处理的模块，各任务之间，既可以独立工作，又可以互相通信。

习 题 一

1. 回答下列问题：

①软件危机的主要表现是什么？

②软件危机产生的原因是什么？

③为什么说不能把Ada看作仅仅是一种普通语言？

④请归纳一下Ada的主要特点。