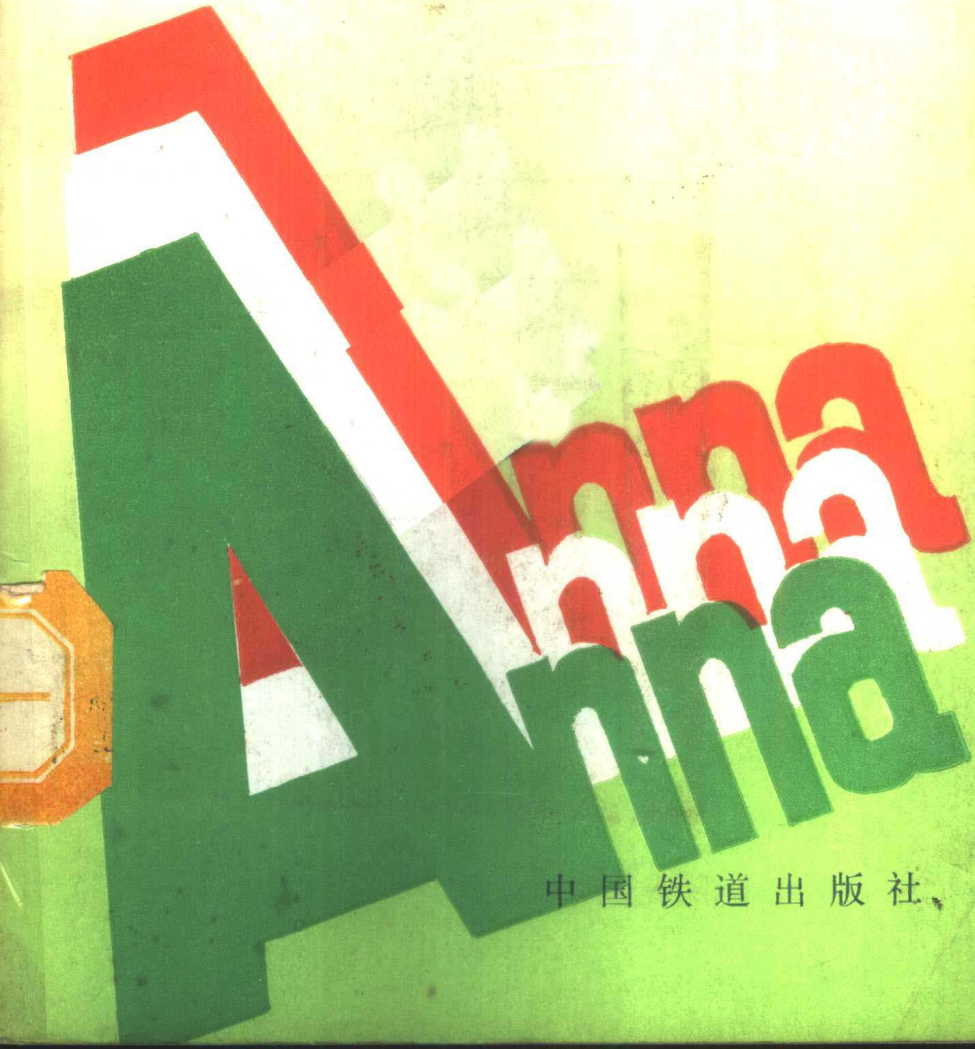
930094

Anna语言导论

Anna yuyan daolun

● 徐宝文 主编



中国铁道出版社

TP312
2830

Anna 语言导论

徐宝文　主编

中国铁道出版社
1990年·北京

内 容 简 介

本书是国内第一本介绍Anna语言的书籍。Anna是一种以Ada为基础的规格说明语言，大量吸收了Pascal等语言的研究成果，可广泛用于软件开发过程中从需求规格说明、设计、编码、测试到维护各个阶段。全书由三部分组成：第一部分是Anna入门，对Anna作一般介绍；第二部分是完整的Anna参考手册；第三部分详细描述了与Anna互补的TSL-1语言。

本书适用于广大软件工作者、高等院校师生，可作为高校本科生与研究生有关课程的教学参考书。

Anna语言导论

徐宝文 主编

*

中国铁道出版社出版、发行

（北京市东单三条14号）

责任编辑 郭 宇 封面设计 刘景山

各地新华书店经销

河北省三河县新燕印刷厂印

开本：787×1092毫米 1/32 印张：7.5 字数：167千

1990年12月 第1版 第1次印刷

印数：000001—1700册

ISBN7-113-00992-1/TP·100 定价：4.15元

前　言

软件形式规格说明技术是软件的一个重要研究课题，但目前可执行的形式规格说明工具还很少见。美国斯坦福大学研制的形式规格说明语言 Anna 是在实用程序设计语言 Ada 的基础上扩充而成的，大量吸收了对 Pascal 等语言的规格说明技术的研究成果。它包括了用于形式地指定 Ada 程序预期行为的设施。设计 Anna 的目的是为了用精确的、机器可处理的标注扩充 Ada，以把有关规格说明与文档编制的有关成果应用到 Ada 软件开发中。Anna 可广泛用于软件开发过程中从需求规格说明、设计、编码、测试到维护各个阶段，是一种既有理论基础又有实用价值的新型形式规格说明语言，它的思想也影响了诸如 PDL 等语言与工具的设计。

本书由三部分组成：第一部分是 Anna 的入门介绍；第二部分是完整的 Anna 参考手册；第三部分详细描述了 TSL-1 语言。TSL-1 是一种 Ada 任务定序语言，也是由美国斯坦福大学的同一研究小组研制的，它弥补了 Anna 目前不能描述并发处理的不足，专用于刻划 Ada 程序的并发行为。因此，Anna 和 Ada 结合起来就可以完整地作 Ada 程序的规格说明。这三部分之后分别附有 Ada、Anna 与 TSL-1 的完整语法描述，读者可以很方便地查阅。

本书是根据美国斯坦福大学计算机系统实验室 D.C.Luckham 教授向编者提供的有关资料的基础上撰写和翻译的，其中：

第一部分《Ada 及其规格说明语言 Anna》与第三部分

《TSL-1——一种Ada任务定序语言》由徐宝文撰写；

第二部分《Anna语言参考手册》由林钧海、秦小麟、蓬颖和徐宝文合作翻译而成：第1章至第3章由蓬颖翻译，第4章第4.13节至第6章由秦小麟翻译，第7章至第12章由林钧海翻译，其余由徐宝文翻译，全部译文由徐宝文校对。

它的出版得到了我院有关领导和同事的关心与帮助并得到中国铁道出版社的大力支持，在此谨致谢意。

由于本书是国内第一本介绍Anna语言的书籍，加之编者水平有限，不足之处定然不少，诚请读者指正。

编　者

1989年岁末于南京航空学院

试读结束：需要全本请在线购买：　www.ertongbook.com

目　录

第一部分：Ada 及其规格说明语言 Anna

第二部分：Anna 语言参考手册

第三部分：TSL-1:一种 Ada 任务定序语言

试读结束：需要全本请在线购买： www.ertongbook.com

第一部分

Ada 及其规格说明语言 Anna

1 引言

众所周知，Ada 是一种功能很强的通用程序设计语言，是高级语言不断发展的产物。尽管 Ada 是在美国国防部主持下针对大型实时嵌入式计算机系统设计的，但由于有世界上众多软件与语言专家参与与关心，由于美国国防部是世界上最大的软件订货商，使得它的影响与应用已渗透到其它许多领域，并会对这些领域起越来越大的作用。

毫无疑问，Ada 是现有过程性语言中功能最强的一个，但是，尽管如此，Ada 还只是一个编码语言，它还不能很好地支撑软件生存期中其它各阶段（虽然有了许多基于 Ada 的 PDL）。而且，即使是在编码阶段，许多信息（尤其是规格说明信息）也很难用 Ada 明确而清楚地表达出来。例如，尽管 Ada 提供了非常丰富的类型与类型定义设施，但我们不可能用它定义一个奇数或偶数类型。虽然我们在定义一个类型时可以用注解指出该类型是奇数或偶数类型，但这种注解的含义是不能由计算机识别的，当一个奇数类型变量的值为偶数时，计算机也不能检查出这种错误。当然，编码者可以在程序中插入必要的检查代码，但这样一来破坏了

程序结构的清晰性，使程序内容主次不分，在有些情况下甚至会出现检查代码比源程序还要长的情况。

为了描述程序的功能需求信息、各种程序成分（如变量、子程序、程序包）的性质及各成分间的相互关系，必须借助另一种工具——规格说明语言。程序员可以用它为程序各部分提供通常不作为程序组成部分的可由计算机处理的精确的规格说明信息，它不仅有助于软件开发的前期阶段，也有助于程序的排错与维护。

美国斯坦福大学研制的 Anna 就是这样一种规格说明语言，它是在 Ada 基础上作必要的扩充后形成的，所扩充的成分主要是一些解释性信息。事实上，这些信息 Ada 本身已可描述一些，Ada 中可以用以描述这类信息的构造有：强类型、子类型、派生类型、程序包、类属单元、异常、上下文子句以及各种约束等。Ada 的这类构造在 Ada 产品的设计中起着双重作用，既可用于提高程序与编译的效率，也可用于提高程序可读性和编译时与运行时的错误检查率，从而减少编程错误。程序员可以用它们显式表示各种程序设计决策，例如一类型的变量的取值范围可以用子类型限定。但是，正如前面所指出的，光有这些结构还是不够的。

我们再从另一方面来看 Ada 的不足。从软件工程的角度，对设计的解释描述应先于实现。但是 Ada 中有些结构由于在设计和实现中过多地考虑到其它因素（如由于强调简单而对结构作了较多的限制），使得这一要求大为减弱。例如，Ada 允许多个任务在不加任何对含义的形式说明的情况下共享全局变量，这样，尽管 Ada 有会合机制用于表述任务间的通信，但程序阅读者不能保证通信就是通过会合机制进行的。

讨论了 Ada 的不足后，就可以有的放矢地对 Ada 进行

扩充。Anna 相对于 Ada 主要扩充增加了一些形式规格说明方面的设施。所作扩充可分为三类：一类是将 Ada 中已有的解释性结构一般化，另一类是增加用于处理异常、上下文子句、子程序的性质声明结构，还有一类是在已有程序规格说明理论研究的基础上增加新的规格说明结构（主要用于指定程序的语义及复合类型与访问类型的运用）。

关于 Anna 的设计目标等参见本书的第二部分。

2 Ada 典型特征

为了帮助读者更准确、更有效地阅读理解下面的内容，这里我们抽出一节对 Ada 基本内容及典型特征作一概述。

Ada 是一种模块化语言，其基本物理模块单元是编译单元，编译单元可以是库单元、库单元体或子单元（前面要加上下文子句）。从逻辑上看，编译单元可以是下述四种程序单元的声明或体：子程序、程序包、任务与类属单元。子程序和其它语言中类似，有过程和函数两种。只是函数调用返回时必须用返回语句返回，对过程两可，子程序及运算符可以重载。

程序包是 Ada 的主要模块设施，一个逻辑程序包可以在物理上分为两部分：规格说明是程序包的可见部分，它列出可由该程序包使用者使用的数据、类型、变量、子程序、任务或其它程序包，用于规定程序包的功能与接口；程序包体是程序包的实现部分，用于描述规格说明中所声明实体的实现细节，这种实现细节使用者不可见，从而也不能使用。

任务是描述并行算法的一种设施，各个任务可分散在多处理机或多计算机上并行执行，也可在一台机器上并发执行。任务的启动是自动进行的，可自动终结也可用中止语句

强行中止．任务在执行过程中可以用延时语句强行挂起．从形式上看，任务类似于程序包，亦由规格说明与体两部分组成，但它与程序包有两点不同，一是后者的规格说明部分可声明任何不含体的实体，而任务的规格说明部分只能声明用于指明通信接口的入口；二是程序包是静态结构而任务是动态结构．任务通信有两种方式，一是通过非局部共享变量进行，二是通过入口调用和接收语句实施．

类属单元是一种用于参数化子程序与程序包的设施，它允许具有相同模式的子程序或程序包共用同一实现细节，这样既可以提高程序的抽象度、减少程序长度，又可以使程序员集中精力优化程序的算法．类属单元在使用之前必须先例举．

Ada 提供了丰富的数据类型，如整数、浮点数、定点数、字符、布尔、字符串、记录、数组等．相应于其它语言中指针类型的是访问类型．它除了提供了一般语言定义类型的方法外，还提供了子类型与派生类型两种定义手段．子类型用于约束其基类型的取值范围，派生类型用于定义同一类类型，以防止不同类型间的隐式转换．Ada 支撑强类型，即类型的属性在编译时就要确定下来，不必作动态类型检查．此外，还可用程序包实现抽象数据类型．

在程序包和类属单元中还可以使用（受限）私有类型．程序包中的私有类型用于防止使用者使用一类型的内部结构及对它不允许的运算，如栈的结构应对其使用者隐藏，使用者不能在栈的非顶部存取元素．类属单元中的私有类型主要用于增大定义模板中类型模式的范围．

语句分简单语句与复合语句．简单语句有赋值语句、过程调用语句、出口语句、返回语句、转移语句、入口调用语句（用于任务通信）、延时语句、中止语句、引发语句（用

于引发异常)、代码语句（用于调用其它语言程序）以及空语句；复合语句有如果语句、分情形语句、循环语句（无限型、步长型、当型）、分程序语句、接收语句与选择语句(这两个语句均用于任务通信与同步，与入口调用语句配合使用)。

Ada 的异常处理机制可以保证程序在正常执行过程中发生异常（如被零除）的情况下能进行相应处理，处理完后程序仍能正常执行。

这里还需要解释一下属性与确立这两个概念。属性指有名实体的性质，例如，枚举类型的上、下界即是该类型的两个属性。属性的计算产生该实体的预定义特性。如设 P 为一变量，则 P'ADDRESS 的计算得到 P 的内存地址。确立指一声明发挥其作用的过程（如对象的建立），确立过程出现在程序执行时。如，对象声明的确立过程如下：首先确立子类型指示或带约束数组定义，从而确立了对象的子类型，然后计算（可能有的）初始化部分，得到对象的初值，再建立对象，最后把初值赋给该对象。实体一经确立就可使用。

为方便使用，Ada 提供了大量的预定义属性与编用(程序员对编译系统的指令）以及由预定义常量、变量、类型、运算符、子程序等组成的预定义程序包。

欲了解 Ada 的详细细节，可参见有关著作及手册，附录中给出了完整的 Ada 语法。

3 Anna 基本概念

Anna 程序是加了形式注解的 Ada 程序。形式注解的语法是通过扩充 Ada 语法范畴并增加新的 Anna 语法范畴来定义的。Anna 中新增加了若干预定义属性、运算与逻辑运

算符，这些新增加的东西只能用在形式注解中。从 Ada 角度看，形式注解就是注解，不影响程序的语义，但在 Anna 程序中，形式注解必须服从 Anna 的语法与语义规则。

Anna 中的形式注解有两种：

- 虚拟 *Ada* 行文，以符号—:开始
- 标注，以符号—|开始。

由此可以看出，任何 Anna 程序都可被标准 Ada 编译程序接收，同时也可为其它 Ada 工具接收，Ada 中是用专用工具处理这些形式注解的。

为与虚拟 Ada 行文相区别，下文中称基本 Ada 程序部分为实际 Ada 行文。所谓基本 Ada 程序指 Anna 程序中除去形式注解后的程序。

3.1 虚拟 Ada 行文

虚拟 Ada 行文是前缀以虚拟注解指示符的 Ada 行文。虚拟行文可以有多种用途。第一个用途是用于定义虚拟概念。程序编得多的人都知道，在程序的构造和解释中往往要用到一些在程序中显式实现的虚拟概念。这些虚拟概念必须精确定义以便用在形式规格说明中。大多数虚拟概念可以用数学函数或谓词表示，在 Anna 中可以用虚拟 Ada 子程序定义虚拟概念。例如，在如下程序包 STACK 中：

```
package STACK  is
—: function LENGTH  return  NATURAL;
   procedure PUSH(X:in ITEM);
—| where in STACK.LENGTH<MAX;
—| out(STACK.LENGTH=in STACK.LENGTH+1);
   function POP  return ITEM;
    ... ...
end STACK;
```


试读结束：需要全本请在线购买： www.ertongbook.com