

GB

1996年 修订—3

中国国家标准汇编

1996 年修订-3

中国标准出版社

1997

图书在版编目 (CIP) 数据

中国国家标准汇编: 1996 年修订-3/中国标准出版社总编室编. —北京: 中国标准出版社, 1997. 10
ISBN 7-5066-1521-5

I. 中… II. 中… III. 国家标准-中国-汇编 IV. T-652
.1

中国版本图书馆 CIP 数据核字 (97) 第 20834 号

中国标准出版社出版
北京复兴门外三里河北街 16 号

邮政编码: 100045

电话: 68522112

中国标准出版社秦皇岛印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

版权专有 不得翻印

*

开本 880×1230 印张 43 $\frac{3}{4}$ 字数 1 392 千字
1997 年 11 月第一版 1997 年 11 月第一次印刷

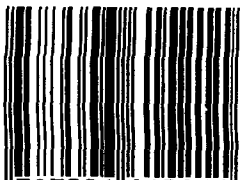
*

印数 1—3 000 定价 120.00 元

*

标目 322—12

ISBN 7-5066-1521-5



787506 615211 >

出版说明

1. 《中国国家标准汇编》是一部大型综合性国家标准全集,自 1983 年起,按国家标准顺序号以精装本、平装本两种装帧形式陆续分册汇编出版。《汇编》在一定程度上反映了我国建国以来标准化事业发展的基本情况和主要成就,是各级标准化管理机构,工矿企事业单位,农林牧副渔系统,科研、设计、教学等部门必不可少的工具书。

2. 由于标准的动态性,每年有相当数量的国家标准被修订,这些国家标准的修订信息无法在已出版的《汇编》中得到反映。为此,自 1995 年起,新增出版在上一年度被修订的国家标准的汇编本。

3. 修订的国家标准汇编本的正书名、版本形式、装帧形式与《中国国家标准汇编》相同,视篇幅分设若干册,但不占总的分册号,仅在封面和书脊上注明“1996 年修订-1,-2,-3,...”等字样,作为对《中国国家标准汇编》的补充。读者配套购买则可收齐前一年新制定和修订的全部国家标准。

4. 修订的国家标准汇编本的各分册中的标准,仍按顺序号由小到大排列(不连续);如有遗漏的,均在当年最后一分册中补齐。

5. 1996 年度发布的修订国家标准分 12 册出版。本分册为“1996 年修订-3”,收入新修订的国家标准 19 项。

中国标准出版社

1997 年 10 月

目 录

GB/T 3057—1996 信息技术 程序设计语言 Fortran	(1)
GB/T 3058—1996 煤中砷的测定方法	(306)
GB 3095—1996 环境空气质量标准	(313)
GB/T 3098.12—1996 紧固件机械性能 螺母锥形保证载荷试验	(319)
GB/T 3098.13—1996 紧固件机械性能 螺栓与螺钉的扭矩试验和破坏扭矩公称直径 1~10mm	(323)
GB/T 3190—1996 变形铝及铝合金化学成分	(328)
GB 3198—1996 工业用纯铝箔	(343)
GB/T 3199—1996 铝及铝合金加工产品包装、标志、运输、贮存	(352)
GB/T 3221—1996 柴油机动力内河船舶系泊和航行试验大纲	(371)
GB 3231—1996 职业性慢性三硝基甲苯中毒诊断标准及处理原则	(441)
GB/T 3295—1996 陶瓷制品 45°镜向光泽度试验方法	(445)
GB/T 3299—1996 日用陶瓷器吸水率测定方法	(449)
GB/T 3366—1996 碳纤维增强塑料纤维体积含量试验方法	(452)
GB/T 3389.1—1996 铁电压电陶瓷词汇	(455)
GB/T 3436—1996 半导体集成电路 运算放大器系列和品种	(481)
GB/T 3477—1996 船用风雨密单扇钢质门	(660)
GB/T 3487—1996 汽车轮辋规格系列	(672)
GB/T 3494—1996 直接法氧化锌	(680)
GB 3531—1996 低温压力容器用低合金钢钢板	(685)

前 言

本标准等同采用国际标准 ISO/IEC 1539:1991《信息技术 程序设计语言 Fortran》和 ISO/IEC 1539:1991/Cor. 1:1993《信息技术 程序设计语言 Fortran 技术修改 1》。

根据 ISO/IEC 1539:1991/Cor. 1:1993, 本标准在 1.4.1、2.4.6、3.3.2.4、4.4.1、5.1、5.1.1.5、5.1.1.7、5.3、7.1.6.1、7.5.3.1、7.5.3.2、9.4.1、9.4.2、10.5.4.1.2、10.8.1、11.3.2、12.1.2.2.1、12.3.2.1、12.3.2.2、12.4.1.1、12.5.2.4、12.5.2.5、12.5.2.8、12.5.4、13.13.13、13.13.25、13.13.66、13.13.80、13.13.100、14.1.3、14.6.2.1 及附录 A 作了改正。

本标准在技术内容上与国际标准完全一致, 仅在如下两点与国际标准略有差异:

- a) 根据 GB 1.1—93 的规定, 表的编号作了改动;
- b) 根据国情附录 E 的内容从略。

本标准说明了以 Fortran 语言所表达的程序的形式, 并建立了对这些程序的解释。它由 Fortran 语言的规格说明所组成。

本标准的附录 A 至附录 F 都是提示的附录。

本标准从生效之日起, 同时代替 GB 3057—82。

本标准由中华人民共和国电子工业部提出。

本标准由电子工业部标准化研究所归口。

本标准起草单位: 中国科学院软件研究所、北京有线电厂、全国信标委程序设计语言分委会。

本标准主要起草人: 程虎、陈明源、吴庆宝、徐祖渊、王懋江、陈天晴、鲍有新、冯惠。

本标准于 1982 年 5 月首次发布。

ISO/IEC 前言

ISO(国际标准化组织)和 IEC(国际电工委员会)是世界性的标准化专门机构。国家成员体(它们都是 ISO 或 IEC 的成员国)通过国际组织建立的各个技术委员会参与制定针对特定技术范围的国际标准。ISO 和 IEC 的各技术委员会在共同感兴趣的领域内进行合作。与 ISO 和 IEC 有联系的其他官方和非官方国际组织也可参与国际标准的制定工作。

对于信息技术,ISO 和 IEC 建立了一个联合技术委员会,即 ISO/IEC JTC1。由联合技术委员会提出的国际标准草案需分发给国家成员体进行表决。发布一项国际标准,至少需要 75%的参与表决的国家成员体投票赞成。

国际标准 ISO/IEC 1539 是由 ISO/IEC JTC1“信息技术”联合技术委员会制定的。

此第二版将取消并代替已作了技术性修改的第一版(ISO 1539:1980)。

附录 A、B、C、D、E 和 F 仅提供参考信息。

引 言

标准程序设计语言 Fortran

本标准说明了以 Fortran 语言(非正式地称作“Fortran90”)所表达的程序的形式,并建立了对这些程序的解释。它由 Fortran 语言的规格说明所组成。在本标准中,未说明此语言的子集。除 1.4.1 中注明的限制外,通常称作“FORTRAN77”的国家标准的语法及语义全部都包含在本标准内。因而,任何遵循标准的 FORTRAN77 程序在本标准下也是遵循标准的。除了本标准文本中指明的少数例外之外,各种新的功能都可以兼容地合并到这类程序中去。

一个遵循标准的 Fortran 处理系统也是一个遵循标准的 FORTRAN77 处理系统。

注意,本语言的名称 Fortran 与 FORTRAN77 的差异在于仅有其第一个字母是大写。FORTRAN77 及 FORTRAN66 在其语言的正式的名称中只使用大写字母,但 Fortran90 并不延续这一传统。

概貌

在本标准对 FORTRAN77 的各项扩充中,下面七项显著的扩充是主要的:

- (1) 数组运算;
- (2) 改善了数值计算设施;
- (3) 参数化的内在数据类型;
- (4) 用户定义的数据类型;
- (5) 用于模块数据定义及过程定义的各种设施;
- (6) 指针;
- (7) 语言进化概念。

在本标准中还包括了其他一些扩充,诸如改进了源程序形式的设施、更多的控制构造、递归、新增的输入/输出设施以及动态可分配数组等。

数组运算

涉及大型数组的计算是工程计算及科学计算的重要部分。在 Fortran 中,数组可以当作实体来使用。由于下述两个主要理由而将用于处理完整数组及子数组(数组片段)的运算包括在本语言中:

(1) 这些功能提供一种更简洁且更高级的语言,它将允许程序员能更迅速且更可靠地开发及维护科学/工程应用系统;

(2) 在许多种计算机体系结构中,这些功能都能极大地方便对数组运算的优化。

FORTRAN77 中的算术运算、逻辑运算、字符运算及内在(预定义的)函数等均被扩展,以便能对数组值操作数进行运算。这类扩展包括完整数组赋值、部分数组赋值及屏蔽的数组赋值、数组值常量及数组值表达式以及用来定义由用户提供的数组值函数的一些设施。还提供了一些新的内在过程,用来操作及构造数组,实现聚集/分散运算,并用来支持与数组有关的扩展的计算能力。例如,提供了一个内在函数,用来对某个数组中的元素求和。

数值计算

科学计算是 Fortran 的基本应用领域之一,而且本标准全部技术工作的指导目标是加强 Fortran 语

言使之成为实现科学领域软件的工具。尽管在科学应用系统中非数值计算急速增长,但数值计算仍处于主导地位。相应的,这些扩充包括:对数值精度说明的可移植控制;对数值内部表示特征的查询;以及改进了的对数值程序性能的控制(例如,改进了的变元范围的减少与求比例)。

参数化的字符数据类型

对于诸如在中国、日本这类使用大字符集的民族语言,通过使用一种用于字符数据类型的种别(Kind)参数而增加了用于多字节字符数据的一些可选设施。这种设施也允许使用用于各种特殊目的的附加字符集,诸如用于数学、化学或音乐等方面的字符。

导出类型

“导出类型”是本标准给予下述这样一组功能的术语:这些功能允许程序员定义一些任意的数据结构并定义对这些数据结构进行的运算。数据结构是用户定义的、且由内在数据类型和导出数据类型构成的聚集。对结构化对象的内在的使用包括赋值、输入/输出以及作为过程变元使用。若不带有由用户定义的、附加的导出类型运算,则该导出数据类型设施是一种简单的数据结构机制。若带有附加的运算定义,则该导出类型为数据抽象提供了一种有效的实现机制。

过程定义可以用于定义对内在类型或导出类型的运算,也可以用于定义对内在类型及导出类型的非内在赋值。

模块定义

在 FORTRAN77 中,无法仅仅在一个地方就定义一个全局数据区域并且使得一个应用系统中的全部程序单元都使用该定义。此外,在实现一组相关的过程(可能涉及公用数据对象)时,ENTRY 语句是不方便的且很受限制。最后,在 FORTRAN77 中没有一种手段使得各个过程定义(尤其是其接口信息)对一个程序单元而言能成为是局部已知的。这些以及其他一些缺陷都由一种新类型的程序单元来弥补,这种程序单元可以包含数据对象声明、导出类型定义、过程定义以及过程接口信息等任意组合。这种称为模块的程序单元可以被认为是块数据程序单元的普遍化及替代品。模块可由任何程序单元所访问,从而使得该模块的内容对该程序单元是可使用的。于是,模块提供了一些改进的设施,用来定义全局数据区域、过程包及封装的数据抽象等。

指针

指针允许数组成为动态大小的及重新排列的,指针还允许结构成为链接的,以便创建表、树及图。一个任意内在类型或导出类型的对象都可以声明为具有指针属性。一旦这样一种对象变为是与一个目标相结合的,它几乎可以出现在一个具有相同类型、相同类型参数及相同形的非指针对象可以出现的任何地方。

语言的变迁

随着一些新的设施的增加,某些旧功能变成冗余的,并由于其用途的衰减,可能最终被淘汰。例如:前面间接提及的数值设施提供了双精度的功能;由于新的数组设施,导致非一致的变元结合(诸如一个数组元素与一个虚拟数组相结合)是不必须的(并且,事实上作为数组运算它是不太有用的);块数据程序单元是冗余的、且劣于模块等。

作为本语言变迁的一部分,提供了一个(已删除的及过时的)语言功能的目录,以便允许本语言中无用的功能从未来的标准中删去。

本标准的构成

本文件由 14 章构成,论及 7 个概念领域。这 7 个领域以及论述此领域的各章如下:

高级/低级概念	第 1、2、3 章
数据概念	第 4、5、6 章
计算	第 7、13 章
执行控制	第 8 章
输入/输出	第 9、10 章
程序单元	第 11、12 章
作用域及结合规则	第 14 章

高级/低级概念

第 2 章(Fortran 术语和概念)里包含了许多 Fortran 的高级概念。它们包括有关一个可执行程序的概念及其主要部分之间的关系。本章还包括了程序单元的语法、语句顺序的规则以及贯穿本文件而被使用的许多基本术语的定义。

第 3 章(字符、词法记号和源程序形式)描述了 Fortran 的低级元素,诸如字符集及允许的源程序形式。其中也包含了为 Fortran 实体构造字面常量及名字的规则,并列出了全部 Fortran 运算符。

数据概念

数组运算(以数组作为数据对象)及数据结构提供了 Fortran 中丰富的数据概念集合。其中主要的概念是数据类型的概念、数据对象的概念以及数据对象的使用的概念,这些概念分别在第 4、5、6 章中描述。

第 4 章(内在的和导出的数据类型)描述了一种数据类型与一个数据对象之间的区别,然后重点描述了数据类型。本章把一种数据类型定义为:各个数据值、表示这些值的相应的各种形式(即各种常量)以及对这些值的各种运算的一个集合。本章也介绍了内在数据类型的概念,并描述了 Fortran 的各种内在类型(INTEGER、REAL、COMPLEX、LOGICAL 及 CHARACTER)的性质。注意,这里只描述了数据对象的类型概念,而没有描述数据对象的声明及其性质的概念。

第 4 章还引入了导出(用户定义的)数据类型的概念,这些类型是一些复合类型,其成分最终将分解为内在类型。给出了定义导出类型的细则(注意:由于内在类型是预定义的,因而不需要也不可能由程序员再定义,因而,内在类型没有对应部分)。与内在类型一样,本章仅仅论及导出类型的类型性质,而不论及导出类型数据对象的声明。

第 5 章(数据对象的声明与说明)详细地描述如何声明有名的数据对象,以及如何给它们以所希望的性质(属性)。数据对象的一个重要属性(也即唯一的对每个数据对象都是必须的属性)是该对象的数据类型,因此,类型声明语句是本章的主要功能。本章详细地描述了各种属性以及说明这些属性的两种方法(类型声明语句及属性说明语句)。本章还描述了隐式地确定类型和存储结合(COMMON 及 EQUIVALENCE),以及数据对象值的初始化。

第 6 章(数据对象的使用)主要论及变量的概念,并描述了变量能具有的各种形式。标量变量包括字符串及子串、结构(导出类型)对象、结构成分以及数组元素。数组及数组片段均被认为是变量。在各种数组设施中,本章描述了数组片段(子数组)以及数组的分配及去分配(用户控制的动态数组)。

计算

第 7 章(表达式和赋值)描述了 Fortran 中如何表达各种计算。这包括了在这些表达式中的表达式

GB/T 3057—1996

操作数(初等量)能具有的形式以及运算符的作用。运算符的优先级用语法规则的形式严格地进行了定义,并以表格形式进行了总结。这章的描述包括了定义的运算符(用户定义的运算符)与内在运算符(+、*、AND、OR、等)的关系。本章详细地描述了对表达式求值的规则以及对内在的运算符及定义的运算符的解释规则(语义)。

第7章还描述了把计算结果赋给数据对象的赋值,它有三种基本的形式:传统的赋值语句,指针赋值语句,以及 WHERE 语句及 WHERE 构造。其中,WHERE 语句及 WHERE 构造允许屏蔽的数组赋值。

第13章(内在过程)描述了一百多个内在过程,它们提供了一组丰富的计算能力。除 FORTRAN77 中已有的内在函数之外,还包括了许多数组处理函数,包括一组完善的数值环境查询函数,以及包括一组以非负整型数据形式来操作位运算的过程。

执行控制

第8章(执行控制)描述了各种控制构造(IF 构造、CASE 构造及 DO 构造等)、各种分支语句(各种形式的 GOTO)以及其他控制语句(IF、算术 IF、CONTINUE、STOP 以及 PAUSE 等)。除了新增加的 CASE 构造以外,以及为包括一个 END DO 终止可选语句、包括几个新增的控制子句以及包括新增的 EXIT 和 CYCLE 语句而扩充的 DO 循环以外,其他语句都与 FORTRAN77 中的相同。

输入/输出

第9章(输入/输出语句)包含了对下述概念的定义:记录;文件;文件连接(OPEN、CLOSE 及预连接文件);包括处理部分记录及变长记录的各种数据传输语句(READ、WRITE 及 PRINT 等);文件定位及文件查询(INQUIRE)。

第10章(输入/输出编辑)描述了输入/输出格式,其中包括了 FORMAT 语句及 FMT=说明符、编辑描述符、表控输入/输出及名表输入/输出等。

程序单元

第11章(程序单元)描述了主程序、外部辅程序、模块以及块数据程序单元。模块连同 USE 语句一起,被描述为封装由其他程序单元使用(访问)的数据和过程定义的一种机制。模块被描述为用来定义下述各项的工具:各种全局的导出类型定义、全局的数据对象声明、过程库以及它们的各种组合。

第12章(过程)包含了对过程(包括用户定义的函数及子程序)的定义及援引的综合论述。本章解释了隐式及显式过程接口的概念,标识出了要求有显式过程接口的各种情况,还描述了管理实元及虚元以及它们的结合的规则。

第12章还描述了在接口块中使用 OPERATOR 可选项,用以允许用中缀运算符和前缀运算符的形式以及用传统的函数形式对函数进行援引。类似地,本章还描述了在接口块中使用 ASSIGNMENT 可选项,以便允许使用子程序调用的一种替代语法。本章还包含了对递归过程、RETURN 语句、ENTRY 语句、内部过程及 CONTAINS 语句、语句函数、类属过程名以及访问非 Fortran 过程的方法等的描述。

作用域及结合的规则

第14章(作用域、结合和定义)解释了术语“作用域”(由于增加了内部过程、模块及其他新功能,现在显得特别重要)的使用,并且描述了各种实体(包括各种名字及各种运算符)的作用域性质;也描述了管理过程变元结合、指针结合以及存储结合的一般规则。最后,第14章还描述了导致变量变为定义(具有可预见的值)的事件以及导致变量变为无定义的事件。

中华人民共和国国家标准

信息技术 程序设计语言 Fortran

GB/T 3057—1996
idt ISO/IEC 1539:1991

Information technology—
Programming language—Fortran

代替 GB 3057—82

1 概述

1.1 范围

本标准规定了以 Fortran 语言表示的程序的形式,并建立以 Fortran 语言表示的程序的解释。本标准的目的是对在各种计算系统上使用的 Fortran 程序增进其可移植性、可靠性、可维护性并有效地执行。

1.2 处理系统

一个计算系统和一种机制的组合,用这个机制来转换程序,使该程序在该计算系统上可以使用,在本标准中就称此组合为处理系统。

1.3 包含和排除

本标准以标识出包含哪些条款和排除哪些条款的方法规定 Fortran 语言的界限。

1.3.1 包含

本标准规定:

- (1)用 Fortran 语言书写的程序可以采用的各种形式;
- (2)解释一个程序及其数据的含义的一些规则;
- (3)由这种程序处理的输入数据的形式;
- (4)由使用这种程序而产生的输出数据的形式。

1.3.2 排除

本标准不规定:

- (1)在计算系统上使用而转换程序的机制;
- (2)为在计算系统上使用程序,而建立和控制此程序所需的操作;
- (3)程序的转换方法,或来自存储介质或到存储介质中去的它们的输入或输出数据的方法;
- (4)除了 1.4 的(2)至(8)项中对处理系统的检测和报表需求外,当本标准的规则对建立一个解释失败时,程序和处理系统的性能;
- (5)程序及其数据的大小或复杂性,它们可能超过任何专用计算系统的容量或特定处理系统的容量;
- (6)在特定处理系统上,量的表示法的物理性质以及数值的舍入、近似和计算的方法;
- (7)输入/输出记录、文件和单元的物理性质;
- (8)存储器的物理性质和执行过程。

1.4 遵循性

本标准中规定的需求、禁止和可选基本涉及到遵循标准的程序能允许的形式和关系,而不涉及处理系统。

若可执行程序(见 2.2.1)只包含在此描述的形式和关系,并且可执行程序有按照本标准的解释,则

该可执行程序是**遵循标准的程序**。若程序单元(见 2.2)能以这样的方式包含在可执行程序中,即允许可执行程序遵循标准,则该程序单元是遵循本标准的。

一个处理系统遵循本标准,若:

(1)它用这样一种方法执行任何遵循标准的程序,它遵循这里的解释,服从任何限制,处理系统可以把这些限制强加在程序的大小和复杂性上。

(2)它包括了检测和报告下述使用的能力,即在所提交的程序单元内使用了这里指明是删除的或过时的形式,这种使用能用编号的语法规则以及和它们相关的约束检测出来。

(3)它包括了检测和报告下述使用的能力,即在提交的程序单元内使用了编号的语法规则以及和它们相关的约束所不允许的附加形式或关系。

(4)它包括了检测和报告下述使用的能力,即在提交的程序单元内使用了处理系统不支持的种别类型参数值(见 4.3)。

(5)它包括了检测和报告下述使用的能力,即在提交的程序单元内使用了第 3 章不允许的源程序形式或字符。

(6)它包括了检测和报告下述使用的能力,即在提交的程序内名字用法的使用与第 14 章中的名字、标号、运算符以及赋值符号的作用域规则不一致。

(7)它包括了检测和报告下述使用的能力,即在提交的程序单元内使用了第 13 章中未定义名字的内在过程。

(8)它包括了检测和报告拒绝一个提交的程序的理由的能力。

然而,在不是 *format-stmt*(见 10.1.1)的一部分的 *format-specification* 中,处理系统不需要检测及报告被删除的功能或过时的功能的使用,或附加形式或关系的使用。

遵循标准的处理系统可以允许附加形式和关系,只要这种附加和标准的形式及关系不相抵触。然而,遵循标准的处理系统可以允许附加的内在过程,即使这样做能引起和遵循标准的程序中的过程名相抵触也没关系。若这样的抵触发生并涉及外部过程名,允许处理系统使用内在过程,除非这个名字被给出一个接口体或被给出同一作用域单元(见第 14 章)中的 EXTERNAL 属性。遵循标准的程序不允许使用由处理系统增加的非标准内在过程。

注意,遵循标准的程序不允许包含国家标准中禁止的任何形式或关系,但若这些形式或关系不改变遵循国家标准的程序的固有解释,则遵循标准的处理系统可以允许这种形式或关系存在,例如,遵循标准的处理系统可以允许另外的内在数据类型。

因为遵循标准的程序可以对处理系统提出超出本标准范围的要求或可以包括非可移植的标准条款,如借助于非 Fortran 定义的外部过程,遵循本标准并不保证遵循标准的程序在所有或任何遵循标准的处理系统上将一致地执行。

在某些情况中,本标准允许某些设施的规定条目,这些设施不完全是在标准中规定的。这些设施被标识为**依赖于处理系统的**,它们必须和由处理系统确定的方法或语义一起向使用者提供。

1.4.1 FORTRAN77 的兼容性

除了本章中注出的之外,本标准是对先前的 Fortran 国家标准 GB3057—82,即非正式地称为 FORTRAN77 的一个向上兼容的扩充,对本标准而言遵循国家标准的处理系统也就是对 FORTRAN77 而言是遵循国家标准的处理系统。任何遵循标准的 FORTRAN77 程序仍然是在本标准之下遵循标准的;然而,应该参见下面(4)项中有关内在过程部分。本标准限制了某些在 FORTRAN77 中依赖于处理系统的功能的行为。所以使用这些依赖于处理系统的功能中的一个功能的这种遵循标准的 FORTRAN77 程序在本标准下就可能有一个不同的解释,但仍然是一个遵循标准的程序。在本标准中下列 FORTRAN77 的性质有不同的解释:

(1)当用常量来初始化 DATA 语句中一个 DOUBLE PRECISION 数据对象时,FORTRAN77 允许处理系统提供比在实型数据中所能包括的精度更精确的精度,而此精度是从实型常量导出的。本标准

不允许处理系统有此可选性。

(2)若一个不在公共块中的有名变量是在 DATA 语句中初始化的且没有规定 SAVE 属性,FORTRAN77 把它的 SAVE 属性留给所依赖的处理系统。本标准规定(见 5.2.9)此有名变量有 SAVE 属性。

(3)FORTRAN77 要求:由输入表要求的字符个数必须小于或等于格式化输入期间在记录中的字符个数。除了在适当的 OPEN 语句中规定 PAD="NO"可选项之外,本标准规定(见 9.4.4.4.2)若在记录中没有足够的字符则用空格逻辑地填充输入记录。

(4)本标准比 FORTRAN77 具有更多的内在函数并加了很少的内在子程序。然而,若遵循标准的 FORTRAN77 程序援引具有和新的标准内在过程中的某个同样名字的过程,则它在本标准之下就有不同的解释,除非在一个 EXTERNAL 语句中将该过程规定为在附录中对非内在函数推荐给 FORTRAN77 标准者之外。

(5)用于格式输出语句中表项的 O 值对一些 G 编辑描述符将有不同形式的格式。此外,Fortran90 标准规定,值如何舍入将影响输出字段形式,但 FORTRAN77 不涉及这个问题;因此,一些 FORTRAN77 处理系统对值和 G 编辑描述符的某些组合将产生与 Fortran90 处理系统不同的输出形式。

(6)FORTRAN77 允许对内在函数进行说明,甚至在对该类型没有定义的函数时亦可。例如

INTRINSIC SIN

LOGICAL SIN

本标准规定(5.1),对内在函数的一个类型声明只可以确认现存的函数的类型。于是,符合标准的 FORTRAN77 程序说明了一个不是为某内在函数定义的类型,将不遵守本标准。

1.5 本标准中使用的记法

在本标准中,“必须”解释为需要,“必须不”解释为禁止。

1.5.1 语法规则

语法规则用来协助描述 Fortran 的词法记号、语句和构造可以采取的形式。这些语法规则以巴科斯-诺尔范式(BNF)的一个变形来表示,其中:

(1)除了注出的之外,来自 Fortran 字符集(见 3.1)的字符要按规定的形式来书写。

(2)小写的斜体字母及词(常用连字符连接并缩写)表示通用语法类,其中特定的语法实体必须以真正的语句代替。在语法术语中使用的某些共同缩写是:

<i>stmt</i>	为语句	(statement)
<i>expr</i>	为表达式	(expression)
<i>spec</i>	为说明符	(specifier)
<i>int</i>	为整数	(integer)
<i>arg</i>	为变元	(argument)
<i>attr</i>	为属性	(attribute)
<i>decl</i>	为声明	(declaration)
<i>def</i>	为定义	(definition)
<i>desc</i>	为描述符	(descriptor)
<i>op</i>	为运算符	(operator)

(3)使用的语法元符号是:

is	引入一个语法类定义
or	引入一个语法类替换
[]	括入一个任选条款
[]...	括入一个任选重复条款,此重复条款可出现零次或多次
■	继续一个语法规则

(4) 每个语法规则都给予形式为 R_{snn} 的一个唯一的标识号, 其中 s 是一位或两位的章号, nn 是该章中的两位顺序号。语法规则适当地分布在正文之中, 并在需要时按标识号引用。第2章和第3章中的某些规则在较后的章节中有更详细的描述; 在这种情况下, 章号 s 是该规则被重复时的较后的那个章号。规则在附录 D 中收集在一起了。

(5) 语法规则不是 Fortran 的完全和精确的语法描述, 并且不能用来自动生成 Fortran 语法分析程序; 在语法规则不完全处, 都附有相应的约束和正文。

(6) 以字形的大小区别来指出过时的功能(见 1.6), 这是对过时的功能使用字形大小的一个例子。

使用语法规则的一个例子是:

digit-string *is* *digit*[*digit*]

以下形式是以上规则所允许的一个数字串形式的例子:

digit

digit digit

digit digit digit digit

digit digit digit digit digit digit digit digit

当对 *digit* 替换以特定的实体时, 真正的数字串可以是:

4

67

1999

10243852

1.5.2 假定的语法规则

为使附加的语法规则个数减至最小以及为传递适当的约束信息, 规定了下列规则。字母“*xyz*”代表任何合法的语法类短语:

xyz-list *is* *xyz*[*,xyz*]....

xyz-name *is* *name*

scalar-xyz *is* *xyz*

约束: *scalar-xyz* 必须是标量。

1.5.3 语法惯例和特征

(1) 任何以“-*stmt*”结束的语法类名遵守源程序形式语句的规则: 它必须由行结束或分号来终止, 若它未形成为另一语句(如 IF 或 WHERE 语句)的一部分, 则它可被标号。反之, 被看作是源程序形式语句的任何东西都在语法规则中给出“-*stmt*”结束。

(2) 语句排列的规则在 *program-unit* (见 R202-R216) 的定义中严格地描述了。表达式层次在 *expr* (见 R723) 的定义中严格地描述了。

(3) 后缀“-*spec*”一般用于说明符, 如输入/输出语句说明符。也用于类型声明属性说明(例如, R512 中的“*array-spec*”), 以及其他很少的情况中。

(4) 当对类型参数(包括括着的圆括号)进行引用时, 使用术语“选择符”。参见“*length-selector*”(见 R507)和“*kind-selector*”(见 R505)。

(5) 术语“*subscript*”(例如 R617、R618 和 R619)一般用于数组定义中。

1.5.4 正文惯例

在描述性正文中, 通常使用与 BNF 语法术语等价的正规英文词。在正文中用大写关键词标识特定的语句和属性, 例如, “END”语句。在正文中也使用黑体字, 在那里以特定的意义对它们进行第一次定义。

1.6 被删除的和过时的功能

本标准保护用户在已有软件中的投资, 这通过包括了所有不依赖于处理系统的 FORTRAN77 的语

(2) 这些功能不包含在 Fortran 的这一版本中。

(6) 只要这些功能继续广泛用于 Fortran 程序, 建议支持 Fortran 语言的处理系统继续支持这些功能。

本标准提供了促进模块和可重用软件的设计和使用的设施。数据和对过程的各种定义可以组织到称为模块的非可执行程序单元内,并使之能用于任何其他程序单元。除全局数据和过程库设施外,模块也对定义数据抽象概念和为某些语言的扩展提供了一种机制。在11.3描述了模块。

可把模块标准化成一个分离的附属标准。**标准模块**必须不使用任何过时的功能,既不使用任何非标准形式也不使用任何非标准关系。

下列标准所包含的条文,通过在本标准中的引用而构成为本标准的条文。本标准出版时,所示版本均为有效。所有的标准都会被修订,使用本标准的各方应探讨使用下列标准最新版本的可能性。

CCIR 建议 460-2

2.1 高级语法

本章介绍了和程序单元有关的术语以及在构造、语句和各表达式级以上的其他 Fortran 概念,并阐明它们之间的关系。1.5.1中描述了语法规则的记法。注意,本章中某些语法规则遵守在较后章条中适当地方给出的约束。

R201 *executable-program* is *program-unit*
[*program-unit*] . . .

一个 *executable-program* 必须精确地包含一个 *main-program program-unit*。

R202 <i>program-unit</i>	is	<i>main-program</i>
	or	<i>external-subprogram</i>
	or	<i>module</i>
	or	<i>block-data</i>

R1101 *main-program* is $\begin{bmatrix} \text{program-stmt} \\ \text{specification-part} \\ \text{execution-part} \end{bmatrix}$

		<i>[internal-sub program-part]</i>
		<i>end-subprogram-stmt</i>
R203 <i>external-subprogram</i>	is	<i>function-subprogram</i>
	or	<i>subroutine-subprogram</i>
R1215 <i>function-subprogram</i>	is	<i>function-stmt</i>
		<i>[specification-part]</i>
		<i>[execution-part]</i>
		<i>[internal-subprogram-part]</i>
		<i>end-function-stmt</i>
R1219 <i>subroutine-subprogram</i>	is	<i>subroutine-stmt</i>
		<i>[specification-part]</i>
		<i>[execution-part]</i>
		<i>[internal-subprogram-part]</i>
		<i>end-subroutine-stmt</i>
R1104 <i>module</i>	is	<i>module-stmt</i>
		<i>[specification-part]</i>
		<i>[module-program-part]</i>
		<i>end-module-stmt</i>
R1110 <i>block-data</i>	is	<i>block-data-stmt</i>
		<i>[specification-part]</i>
		<i>end-block-data-stmt</i>
R204 <i>specification-part</i>	is	<i>[use-stmt]. . .</i>
		<i>[implicit-part]</i>
		<i>[declaration-construct]. . .</i>
R205 <i>implicit-part</i>	is	<i>[implicit-part-stmt]. . .</i>
		<i>implicit-stmt</i>
R206 <i>implicit-part-stmt</i>	is	<i>implicit-stmt</i>
	or	<i>parameter-stmt</i>
	or	<i>format-stmt</i>
	or	<i>entry-stmt</i>
R207 <i>declaration-construct</i>	is	<i>derived-type-def</i>
	or	<i>interface-block</i>
	or	<i>type-declaration-stmt</i>
	or	<i>declaration-stmt</i>
	or	<i>parameter-stmt</i>
	or	<i>format-stmt</i>
	or	<i>entry-stmt</i>
	or	<i>stmt-function-stmt</i>
R208 <i>execution-part</i>	is	<i>executable-construct</i>
		<i>[execution-part-construct]. . .</i>
R209 <i>execution-part-construct</i>	is	<i>executable-construct</i>
	or	<i>format-stmt</i>
	or	<i>data-stmt</i>
	or	<i>entry-stmt</i>
R210 <i>internal-subprogram-part</i>	is	<i>contains-stmt</i>
		<i>internal-subprogram</i>
		<i>[internal-subprogram]. . .</i>