

高级 Fortran 笔记

GasinAn

2024 年 4 月 4 日

Copyright © 2024 by GasinAn

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher, except by a L^AT_EXer.

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories, technologies and programs to determine their effectiveness. The author and publisher make no warranty of any kind, express or implied, with regard to these techniques or programs contained in this book. The author and publisher shall not be liable in any event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these techniques or programs.

Printed in China

目录

第一章 Fortran 概念	1	2.3.3 TYPE 类型说明符	8
1.1 程序单元概念	1	2.3.4 CLASS 类型说明符	8
1.1.1 程序单元和用域单元	1	2.4 派生类型	9
1.1.2 程序	1	2.4.1 派生类型概念	9
1.1.3 过程	1	2.4.2 派生类型定义	9
1.2 数据概念	1	2.4.3 派生类型参数	10
1.2.1 类型	1	2.4.4 成员	10
1.2.2 数据值	2	第三章 属性声明和说明	15
1.2.3 数据实体	2	3.1 过程和数据对象的属性	15
1.2.4 引用	3	3.2 类型声明语句	15
1.2.5 数组	3	3.3 自动数据对象	16
1.2.6 可分配变量	4	3.4 属性	16
第二章 类型	5	3.4.1 ALLOCATABLE 属性	16
2.1 类型的特征	5	3.4.2 EXTERNAL 属性	17
2.1.1 类型的概念	5	3.4.3 POINTER 属性	17
2.1.2 类型分类	5	3.4.4 VALUE 属性	17
2.1.3 值的集合	5	3.4.5 VOLATILE 属性	17
2.1.4 常量	5		
2.2 类型参数	6		
2.3 类型, 类型说明符, 和值	7		
2.3.1 类型与值和对象的关系	7		
2.3.2 类型说明符和类型匹配	7		

第一章 Fortran 概念

1.1 程序单元概念

1.1.1 程序单元和作用域单元

程序单元是 Fortran 程序的基本组件. 程序单元是主程序, 外部子程序, 模块, 或子模块.

子程序是函数子程序或子例行子程序.

1.1.2 程序

一个程序应包含一个主程序, 大于等于零个其他种类的程序单元, 大于等于零个外部过程, 和大于等于零个以 Fortran 之外的方式定义的实体.

外部过程是以外部子程序或 Fortran 之外的方式定义的过程.

1.1.3 过程

过程是函数或子例行.

1.2 数据概念

1.2.1 类型

总述

类型是被命名的数据分类, 和其类型参数一起决定值的集合, 表示值的语法, 和解释与操作值的运算的集合.

类型是固有类型或派生类型.

固有类型

固有类型是整型, 实型, 复型, 字符型和逻辑型.

所有固有类型都有一个种别类型参数称作 `KIND`, 其决定相应类型的表示方法. 字符型还有一个长度类型参数称作 `LEN`, 其决定字符串的长度.

派生类型

派生类型可以被参数化. 派生类型的标量对象是结构体; 结构体的赋值已被固有地定义, 但没有结构体的固有运算. 对每个派生类型, 都有一个结构体构造器可用于生成值. 另外, 派生类型的对象可用成过程参量和函数结果, 并且可以出现于输入/输出列表. 如果另外的运算被派生类型所需要, 则可以由某些过程定义.

1.2.2 数据值

根据类型参数的值, 每个固有类型都与彼类型的数据可以取的值的集合联系. 派生类型的对象可假定的值决定于类型定义, 类型参数值, 和其成员的值的集合.

1.2.3 数据实体

总述

数据实体是数据对象, 表达式的求值的结果, 或函数引用的执行的结果.

一个数据实体有一个类型和类型参数, 并且可能有一个数据值 (一个例外是未定义变量). 每个数据实体都有一个维数并因此要么是一个标量要么是一个数组.

是函数引用的执行的结果的数据实体被称为函数结果.

数据对象

数据对象分类 数据对象要么是常量, 要么是变量, 要么是常量的子对象. 具名数据对象的类型和类型参数可以被显式地或隐式地说明.

子对象是可独立于其他部分被引用的数据对象的一部分. 如果子对象是变量, 则还需要可独立于其他部分被定义.

这些“部分”包括数组的部分 (数组元素和数组片段), 字符串的部分 (子串), 复型对象的部分 (实部和虚部), 和结构体的部分 (成员). 子对象本身是数据对象, 但子对象只由对象指示符或固有函数引用. 变量的子对象是变量.

变量 变量可以有一个值或是未定义变量; 在程序执行期间可以被定义, 重定义, 或重新变得未定义.

常量 常量要么是具名常量要么是字面常量.
具名常量用 `PARAMETER` 属性定义.

常量的子对象 常量的子对象是常量的一部分.

在常量的子对象的对象指示符中, 被引用的部分可能取决于某个变量的值. (示例: 在数组片段中使用整型变量做索引)

表达式 表达式在被求值时生成一个数据实体. 表达式要么表示一个数据对象引用要么表示一个计算, 并由运算对象, 运算符和括号组成.

函数引用 函数引用在函数于表达式求值期间被执行时生成一个数据实体. 函数结果的类型, 类型参数, 和 维数由函数的接口接口决定. 函数结果的值由函数的执行决定.

1.2.4 引用

数据对象在其值于执行期间被请求时被引用. 过程在其被执行时被引用.

数据对象指示符或过程指示符作为实参量的出现并不构成对该数据对象或过程的引用, 除非这种引用对于完成实参量的说明是必要的.

1.2.5 数组

一个数组可以有最多十五减其协秩个维度, 和任意长度在其任意维度上. 一个数组的大小是其所有元素的总数, 等于所有长度的乘积. 一个数组可以是零大小的. 一个数组的形状决定于其维数和其每个维度上的长度, 并表示为一个元素是其所有长度的 1 维数数组. 所有具名数组必须被声明, 并

且具名数组的维数在其声明中被说明. 除假定秩数组外, 一旦被声明, 具名数组的维数就是常量.

为标量对象定义的任何固有运算都可以应用于一致对象. 这样的运算被逐元地进行以生成一个和数组的运算对象一致的作为结果的数组. 如果一个逐元运算是固有纯粹运算或由一个纯粹逐元函数实现, 那么这个逐元运算可以被同时地进行或以其他顺序进行.

一个 1 维数数组可由标量和其他数组构造并且可以被变形成任意允许的数组形状.

1.2.6 可分配变量

可分配变量的分配状态要么是已分配要么是未分配.

一个未分配变量不应被引用或定义.

如果一个可分配变量是一个数组, 那么其维数已被声明, 但其界在其是已分配的时候才是确定的.

第二章 类型

2.1 类型的特征

2.1.1 类型的概念

Fortran 提供了一种抽象的方法, 通过这种方法可以在不依赖于特定物理表现的情况下对数据进行分类. 这种抽象的方法是类型的概念.

一个类型有一个名称, 一个有效值的集合, 一个表示这些值 (常量) 的方法, 和一个操作这些值的运算的集合.

2.1.2 类型分类

类型要么是固有类型要么是派生类型.

派生类型是由派生类型定义或固有模块定义的, 并且应该仅当是可访问之时被使用. 固有类型总是可访问.

2.1.3 值的集合

对每个类型, 都有一个有效值的集合. 逻辑型的有效值的集合不是处理器依赖的. 整型, 字符型, 和实型的有效值的集合是处理器依赖的. 复型的有效值的集合由所有实部和虚部的值的结合的集合组成.

2.1.4 常量

表示值的语法表明类型, 类型参数, 和特定的值.

是常量表达式的结构体构造器表示派生类型的标量常量值. 是常量表达式的数组构造器表示固有类型或派生类型的常量数组值.

常量值可以具名.

对每个固有类型, 一个运算和相应的运算符的集合以被固有地定义. 这固有集合可以由通过带 OPERATOR 接口的函数定义的运算和运算符扩充.

对派生类型, 没有固有运算. 派生类型上的运算可由程序定义.

2.2 类型参数

如果类型有类型参数, 那么值的集合, 表示值的语法, 和类型的值上的运算的集合取决于参数的值.

类型参数要么是种别类型参数要么是长度类型参数. 所有类型参数都是整型的. 种别类型参数参与泛型解析, 但长度类型参数不参与.

每个固有类型都有一个名为 KIND 的种别类型参数. 固有类型字符型一个名为 LEN 的长度类型参数. 派生类型可以有类型参数.

类型参数值可由类型说明说明.

类型参数值 是 标量整型表达式
或 *
或 :

种别类型参数的类型参数值应当是常量表达式.

冒号不应被用作类型参数值除非在有 POINTER 或 ALLOCATABLE 属性的实体的声明中.

作为类型参数值的冒号说明延迟类型参数.

对象的延迟类型参数的值决定于 ALLOCATE 语句的成功执行, 固有赋值语句的执行, 指针赋值语句的执行, 或参量关联.

作为类型参数值的星号说明长度类型参数是假定类型参数. 其被哑参量用于从有效参量中假定类型参数值, 被 SELECT TYPE 结构中的关联名称用于从相关的选择器中假定类型参数值, , 且被字符型的具名常量用于从常量表达式中假定字符长度.

种别类型参数的值永远在编译时刻已知.

“长度类型参数”的名称被用于非种别类型参数的类型参数是因为它们经常说明长度. 然而, 它们也可被用于其他用途. 它们和种别类型参数的重要区别是它们的值无需在编译时刻已知并且在执行时可能变化.

2.3 类型, 类型说明符, 和值

2.3.1 类型与值和对象的关系

类型的名称充当类型说明符并可用于声明该类型的对象. 声明可以说明具名对象的类型. 数据对象可以显式地或隐式地声明. 数据对象除了其类型之外还有属性.

数组由固有类型或派生类型的标量数据组成, 并具有与其元素相同的类型和类型参数.

变量是数据对象. 变量的类型和类型参数决定该变量可以取哪些值. 赋值提供了一种改变变量的值的方法.

变量的类型决定可用于操作该变量的运算.

2.3.2 类型说明符和类型匹配

类型说明符说明类型和类型参数值. 其要么是类型说明符要么是声明类型说明符.

类型说明符 是 固有类型说明符
或 派生类型说明符

派生类型说明符不应说明抽象类型.

声明类型说明符 是 固有类型说明符
或 TYPE(固有类型说明符)
或 TYPE(派生类型说明符)
或 CLASS(派生类型说明符)
或 CLASS(*)
或 TYPE(*)

在声明类型说明符中, 每个不是冒号或星号的类型参数值都应该说明表达式.

在使用 CLASS 关键字的声明类型说明符中, 派生类型说明符应该说明一个可扩展类型.

TYPE(派生类型说明符) 不应说明抽象类型.

在 TYPE(固有类型说明符) 中, 固有类型说明符不能以逗号结尾.

用 CLASS 关键字声明的实体应该是一个哑参量或具有 ALLOCATABLE 或 POINTER 属性.

注: TYPE 类型说明符和 CLASS 类型说明符的区别在于 TYPE 类型

说明符只代表某一类型而 CLASS 类型说明符代表某一类型及其所有子类型.

2.3.3 TYPE 类型说明符

TYPE 类型说明符用于声明假定类型实体, 或固有类型或派生类型实体.

类型声明语句中的 TYPE 类型说明符中的派生类型说明符应当说明先前被定义的派生类型. 如果数据实体是函数结果, 则只要派生类型是在函数体中被定义的或在函数体中通过使用关联或宿主关联是可访问派生类型就可在 FUNCTION 语句 中被说明. 如果派生类型在 FUNCTION 语句中被说明并在函数体中被定义, 那么这就好像函数结果紧接着被说明的派生类型的派生类型定义用彼派生类型被声明.

2.3.4 CLASS 类型说明符

CLASS 类型说明符用于声明多态实体. 多态实体是在程序执行期间能够具有不同动态类型的数据实体. 动态类型是数据实体在程序执行期间的特定点的类型.

类型声明语句中的 CLASS 类型说明符中的派生类型说明符应当说明先前被定义的派生类型. 如果数据实体是函数结果, 则只要派生类型是在函数体中被定义的或在函数体中通过使用关联或宿主关联是可访问派生类型就可在 FUNCTION 语句 中被说明. 如果派生类型在 FUNCTION 语句中被说明并在函数体中被定义, 那么这就好像函数结果紧接着彼派生类型的派生类型定义用彼派生类型被声明.

如果 CLASS 类型说明符包含类型名称, 那么多态实体的被声明类型为被说明的类型. 被声明类型是数据实体被要么显式地要么隐式地声明具有的类型.

非多态实体仅和相同被声明类型的实体有类型匹配. 不是无限多态实体的多态实体和相同被声明类型的或其扩展的实体有类型匹配. 如果实体与某类型的实体有类型匹配那么其与该类型有类型匹配.

2.4 派生类型

2.4.1 派生类型概念

另外的类型可以从固有类型和其他派生类型派生出来. 类型定义定义类型的名称及其成员和类型绑定过程的名称和属性.

派生类型可以通过一个或多个类型参数进行参数化, 每个类型参数都要么被定义为种别类型参数要么被定义为长度类型参数并且可以具有默认值.

派生类型的根本成员是固有类型或具有 ALLOCATABLE 属性或 POINTER 属性的成员, 加上派生类型的既没有 ALLOCATABLE 属性也没有 POINTER 属性的成员的根本成员.

派生类型的直接成员是该类型的成员, 加上既没有 ALLOCATABLE 属性也没有 POINTER 属性的派生类型的成员的直接成员.

派生类型的潜在子对象成员是该类型的非指针成员以及派生类型的非指针成员的潜在子对象成员. 这包括所有可以是该类型的对象的子对象的成员.

派生类型的对象的成员, 直接成员, 潜在子对象成员和根本成员分别是其类型的成员, 直接成员, 潜在子对象成员和根本成员.

默认情况下, 成员定义的顺序不暗指存储序列. 然而, 一个序列类型暗指一个存储序列.

派生类型的标量实体是一个结构体. 如果派生类型具有 SEQUENCE 属性, 则该类型的标量实体是一个序列结构体.

2.4.2 派生类型定义

派生类型定义的语法

```
派生类型定义  是  派生类型语句
                  [类型参数定义语句]
                  [私有或序列]
                  [成员部分]
                  [类型绑定过程部分]
                  结束类型语句

派生类型语句  是  TYPE[[, 类型属性说明列表] ::] 类型名称 [(类型参数名称列表)]
```

类型属性说明 是 ABSTRACT
或 访问说明
或 BIND(C)
或 EXTENDS(父类型名称)

派生类型类型名称不应是 DOUBLEPRECISION 或和任何固有类型的名称相同.

如果类型定义包含或继承一个延迟类型绑定过程, ABSTRACT 应当出现.

如果 ABSTRACT 出现, 那么类型应当可扩展.

如果 EXTENDS 出现, 那么 SEQUENCE 不应当出现.

私有或序列 是 私有成员语句
或 序列语句

2.4.3 派生类型参数

类型参数定义语句

类型参数定义语句 是 整型, 类型参数属性说明 :: 类型参数声明列表
类型参数声明 是 类型参数名称 [= 标量整型常量表达式]
类型参数属性说明 是 KIND
或 LEN

类型参数顺序

类型参数顺序是派生类型的类型参数的顺序; 它被为派生类型说明器使用.

非扩展后类型的类型参数顺序是派生类型定义中的类型参数名称列表中的顺序. 扩展后类型的类型参数顺序由其父类型的类型参数顺序按派生类型定义中的类型参数名称列表的顺序后跟所有另外的类型参数组成.

2.4.4 成员

成员定义语句

成员部分 是 [成员定义语句] ...
成员定义语句 是 数据成员定义语句
或 过程成员定义语句

过程成员定义语句 是 PROCEDURE([过程接口]), 过程成员属性说明器列表:: 过程声明列表

过程成员属性说明器 是 访问说明器

或 NOPASS

或 PASS[(参量名称)]

或 POINTER

在给定的过程成员定义语句中相同的过程成员属性说明器不应出现多于一次.

在每个过程成员属性说明器列表中 POINTER 都应出现.

如果过程指针成员有一个隐式接口或没有参量, NOPASS 应被说明.

如果 PASS(参量名称) 出现, 过程指针成员的接口应有一个名称为参量名称的哑参量.

在相同的过程成员属性说明器列表中 PASS 和 NOPASS 不应都出现.

被传递对象哑参量

被传递对象哑参量是类型绑定过程或过程指针成员的与过程由此被激活的对象关联的哑参量.

被传递对象哑参量是过程指针成员或类型绑定过程的一个被区分的哑参量. 它影响过程重写和参量关联.

如果 NOPASS 被说明, 那么过程指针成员或类型绑定过程没有被传递对象哑参量.

如果 PASS 和 NOPASS 都没被说明或 PASS 被不带参量名称地说明, 那么过程指针成员或类型绑定过程的第一个哑参量是其被传递对象哑参量.

如果 PASS(参量名称) 被说明, 那么名称为参量名称的哑参量是过程指针成员或具名类型绑定过程的被传递对象哑参量.

被传递对象哑参量应是与被定义的类型相同的被声明类型被声明类型的标量非指针非可分配哑数据对象; 其所有的长度类型参数都应被假定; 其应多态当且仅当被定义的类型是可扩展类型. 其不应有 VALUE 属性.

成员顺序

类型参数顺序是派生类型的类型参数的顺序; 它被为派生类型说明器使用.

非扩展后类型的成员顺序是派生类型定义中的成员的声明的顺序. 扩展后类型的成员顺序由其父类型的成员顺序按派生类型定义中的成员的声明

的顺序后跟所有另外的成员组成.

类型绑定过程

类型绑定过程部分 是 包含语句
[绑定私有语句]
[类型绑定过程绑定]

绑定私有语句 是 PRIVATE

仅当类型定义在模块的说明部分内时绑定私有语句是被允许的.

类型绑定过程绑定 是 类型绑定过程语句
或 类型绑定泛型语句
或 终止过程语句

类型绑定过程语句 是 PROCEDURE[[, 绑定属性列表] ::] 类型绑定过程声明列表
或 PROCEDURE(接口名称), 绑定属性列表:: 绑定名称列表

类型绑定过程声明 是 绑定名称 [=> 过程名称]

如果 => 过程名称不在类型绑定过程声明中出现, 那么过程名称和绑定名称相同.

类型绑定泛型语句 是 GENERIC[, 访问说明] :: 泛型说明 => 绑定名称列表

绑定属性 是 访问说明
或 DEFERRED
或 NON_OVERRIDABLE
或 NOPASS
或 PASS(参量名称)

如果类型绑定过程或终止子例行的接口没有被定义的类型参量, NOPASS 应出现.

如果 PASS(参量名称) 出现, 类型绑定过程或终止子例行的接口应有一个名称为参量名称的哑参量.

在相同的绑定属性列表中 PASS 和 NOPASS 不应都出现.

在相同的绑定属性列表中 DEFERRED 和 NON_OVERRIDABLE 不应都出现.

当且仅当接口名称出现时 DEFERRED 应当出现.

带 DEFERRED 属性的类型绑定过程是延迟类型绑定过程. DEFERRED 关键字只应出现在抽象类型的定义中.

终止子例行

终止过程语句 是 FINAL[::] 终止子例行名称列表

终止子例行名称应是带且只带一个哑参量的模块过程的名称. 彼参量应是被定义的派生类型的非可选非协数组非指针非可分配非多态变量. 所有长度类型参数应被假定. 哑参量应不有 INTENT(OUT) 或 VALUE 属性

终止子例行不应有和彼类型的另一个终止子例行的哑参量有相同的种别类型参数和维数的哑参量.

如果一个终止子例行有一个假定秩数组哑参量, 那么彼类型的其他终止子例行不应有和此假定秩数组哑参量有相同种别类型参数的哑参量.

终止子例行不通过类型扩展被继承且不可被重写. 父类型的终止子例行在扩展后类型所有另外的终止子例行被调用后被调用.

第三章 属性声明和说明

3.1 过程和数据对象的属性

每个数据对象都有一个类型和维数并且可以有类型参数和其他决定对象的用处的特性. 合起来, 这些特性是对象的属性.

函数有一个类型和维数并且可以有类型参数和其他决定函数的用处的属性. 类型, 维数和类型参数与函数结果的相同.

子例行没有类型, 维数或类型参数, 但可以有其他决定子例行的用处的属性.

3.2 类型声明语句

类型声明语句 是 声明类型说明 [[, 属性说明] ... ::] 实体声明列表

属性说明 是 ASYNCHRONOUS
 或 BIND(C[, NAME= 标量默认字符常量表达式])
 或 CODIMENSION[协数组说明]
 或 CONTIGUOUS
 或 DIMENSION[数组说明]
 或 EXTERNAL
 或 INTENT(目的说明)
 或 INTRINSIC
 或 OPTIONAL
 或 PARAMETER
 或 POINTER
 或 PRIVATE
 或 PROTECTED
 或 PUBLIC
 或 SAVE
 或 TARGET
 或 VALUE
 或 VOLATILE

如果 PARAMETER 关键字出现, 那么初始化应当在每个实体声明中出现.

3.3 自动数据对象

自动数据对象是有决定于不是常量表达式的说明表达式的值的类型参数或数组界的非哑数据对象.

自动数据对象不应有 SAVE 属性.

3.4 属性

3.4.1 ALLOCATABLE 属性

只有变量和成员能有 ALLOCATABLE 属性. 引用一个结果变量有 ALLOCATABLE 属性的函数的结果是本身没有 ALLOCATABLE 属性的一个值.

3.4.2 EXTERNAL 属性

如果外部过程或哑过程被用作实参量或是过程指针赋值的目标, 那么它应被声明有 EXTERNAL 属性.

EXTERNAL 属性可在类型声明语句中, 通过接口体, 通过 EXTERNAL 语句, 或通过过程声明语句被说明.

3.4.3 POINTER 属性

带 POINTER 属性的过程应有 EXTERNAL 属性.

3.4.4 VALUE 属性

VALUE 属性说明哑参量的参量关联的一种类型.

带 VALUE 属性的实体应是哑参量对象. 它不应是假定大小数组, 协数组或带协数组根本成员的变量.

带 VALUE 属性的实体不应有 ALLOCATABLE , INTENT(INOUT) , INTENT(OUT) , POINTER , 或 VOLATILE 属性.

过程的带 BIND 属性的哑参量不应既有 OPTIONA 属性又有 VALUE 属性.

如果 VALUE 属性被说明, 效果是如同实参量被赋值给一个临时变量, 并且彼变量随后被同哑参量参量关联. 结果是哑参量的值或定义状态的改变不影响实参量.

3.4.5 VOLATILE 属性

VOLATILE 属性说明对象可以不由程序说明的方式被引用, 定义, 或成为未定义.

