2023-10-31 Ver 1.0

FortranWG 岩下

J3/23-155r2 の解説

1. はじめに

23-155 は Fortran の拡張となるジェネリックプログラミング機能の文法を提案しています。2 章は提案機能の文法をほぼ網羅した例です。3 章に文法が書かれています。4 章はその他の変更箇所です。

2. 例

- TEMPLATE 構文 TEMPLATE B(T,F,C) 〜 END TEMPLATE B は，テンプレートを定義しています。
- TEMPLATE 構文の宣言部では，引数 T, F 及び C について次のように宣言しています。
  - REQUIRES 文は，同じ名前をもつ REQUIREMENT 構文と対で使用されます。ここでは，テンプレート引数のうち T と F を REQUIREMENT 構文に受け渡し，型 T と関数 F の関係を宣言しています。
  - TYPE,DEFERRED で始まる文（deferred 型宣言文）は，引数 T が任意の型を持つことを宣言しています。
  - INTEGER,CONSTANT で始まる文（deferred 定数宣言文）は，引数 C が整定数であることを宣言しています。
- TEMPLATE 構文の内部のサブルーチン SUB1 と SUB2 は，型 T と関数 F を使って定義されています。T と F は，テンプレートが具体化されるときに確定します。
- INSTANTIATE 文は，テンプレートに引数を与えて具体化します。第 1 の INSTANTIATE 文は，TEMPLATE 構文中の T を REAL 型に，F を演算子 * に，C を 1 次元整定数配列 [3, 4] に置き換えたインスタンスを生成します。その結果，SUB1 と SUB2 は使用できるサブプログラムとなります。
- INSTANTIATE 文では，'=>' 記法によってテンプレート中の名前に別名を付けることができます。これによって，名前の衝突を回避することができます[1]。

[1] INSTSANTIATE 文中の tot_sub1, max_sub1 は，それぞれ tot_sub, max_sub の間違いだと思われます。

3. 構文

3.1 Deferred 言語要素

REQUIREMENT 構文と TEMPLATE 構文の引数を，deferred 引数と呼びます。deferred とは，その全体または一部が抽象化されていて，INSTANTIATE 文による具体化まで確定が遅延されるという意味ですが，提案書全体を通して重要な用語なので，訳さずにそのまま使うことにします。deferred 引数には，以下の 3 種類があります。

- deferred 定数は，値が抽象化された定数です(3.1.1.1)。
  - 型と形状は，deferred 定数宣言文 (deferred-const-declaration-stmt) によって指定します。
  - その型は，整数型，論理型，または文字型でなければならず，文字型のとき文字長引継ぎでなければなりません。
- deferred 手続は，その引用仕様以外の定義が抽象化された手続です(3.1.1.2)。
  - それは，deferred 引数宣言文 (deferred-arg-declaration-stmt) 中に現れるか，引用仕様本体 (interface body) または deferred 引用仕様本体 (deferred-interface-body) の関数名またはサブルーチン名として現れなければなりません。
  - その引用仕様は，引用仕様または deferred 引用仕様本体によって指定されます。
- deferred 引用仕様本体は，引用仕様宣言 (inetrface block) 中の引用仕様本体とほぼ同じ形をしていますが，親子結合 (host association) が適用されます(3.1.2)[2]。
  - deferred 型は，その全てが抽象化された型です(3.1.1.3)。
  - deferred 型宣言文 (deferred-type-declaration-stmt) によって宣言されなければなりません。
  - deferred 型は，多相的にはなれません。

全ての deferred 引数は，TEMPLATE 構文中または REQUIREMENT 構文中に宣言が必要です。同じ deferred 引数に対して二つ以上の宣言があるとき，それらの間には一貫性がなければなりません(3.1.3)。

- deferred 定数(3.1.3.1)
  - その型は，指定された型（整数型，実数型などの組込み型，または派生型）となります。型の指定は省略できません。
  - その型パラメタは，指定があればその値となり，なければ deferred となります。
  - その形状（次元数と各次元の大きさ）は，形状を明示する宣言があればその形状となり，なければ形状引継ぎ配列となって，実引数から形状を引き継ぎます。
- deferred 手続(3.1.3.2)
  - サブルーチンまたは関数のどちらか一つとして宣言されます。

---

[2] deferred 引用仕様本体内では，親スコープの言語要素（例えば deferred 型）が参照できます。引用仕様宣言中の引用仕様本体では，IMPORT 文による指定がない限り，親スコープの言語要素は参照できません。

- 単純手続 (simple procedure) または純粋手続 (pure procedure) であることの指定は，少なくとも一つの宣言にあれば有効です。
- 要素別処理手続 (elemental procedure) であることの指定は，少なくとも一つの宣言にあれば有効です。

INSTANTIATE 文と TEMPLATE 構文との間の引数結合のルールは，引数の並び順やキーワード引数の使い方については，従来の手続呼出しにおけるルールと同じです(3.1.4)。加えて，引数の種類によって次のルールがあります。

- 定数(3.1.4.1)
  - 実引数は，整数型，論理型，または文字型の定数（次元数は任意）です。
  - 実引数は，deferred 定数である仮引数に対応しなければなりません。
  - 実引数は，仮引数と同じ型と種別を持たなければなりません。
  - 仮引数が形状引継ぎ配列でないとき，実引数は仮引数と同じ形状をもたなければなりません。
  - 仮引数が次元数引継ぎ配列でないとき，実引数は仮引数と同じ次元数をもたなければなりません。
- 手続(3.1.4.2)
  - 実引数は，総称指定（総称名，演算子など）または手続名です。
  - 実引数は，deferred 手続である仮引数に対応しなければなりません。
  - 手続名である実引数は，仮引数と同じ性質をもたなければなりません。ただし，PURE 属性，SIMPLE 属性及び ELEMENTAL 属性については，もし仮引数がそのいずれかをもっていても，実引数は必ずしも同じ属性をもつ必要はありません。
  - 総称指定である実引数は，その一つの個別手続が，仮引数と同じ性質をもたなければなりません。ただし，PURE 属性，SIMPLE 属性及び ELEMENTAL 属性については，もし仮引数がそのいずれかをもっていても，個別手続は必ずしも同じ属性をもつ必要はありません。この対応付けによって，総称指定の候補の中から個別手続が選択されます。
- 型(3.1.4.3)
  - 実引数は，deferred 型である仮引数に対応しなければなりません。
  - その他，複雑な条件の制限が 2 件と，共配列に関わる制限が 1 件あります。

3.2 REQUIREMENT 構文

REQUIREMENT 構文は，deferred 引数の宣言の集まりであり，名前をもちます。REQUIREMENT 構文を使えば，TEMPLATE 構文や他の REQUIREMENT 構文の中で繰り返されるパターンを再利用することができます。

REQUIREMENT 構文には，次の構文だけを書くことができます。

- USE 文
- deferred 定数宣言文，deferred 手続宣言文，deferred 型宣言文 (3.1.1)
- REQUIRES 文 (3.3)
- deferred 引用仕様本体 (3.1.2)

REQIREMENT 構文は，参照結合と親子結合が許される有効域です。

3.3 REQUIRES 文

REQUIRES 文には，REQUIREMENT 構文の名前と，REQUIREMENT 構文の deferred 引数に対応付ける deferred 引数を指定します。同じ名前をもつ REQUIREMENT 構文は，事前に定義されていなければなりません。

3.4 TEMPLATE 構文

テンプレートは，インスタンス化 (instantiation) によって有効になる宣言と定義の集合です。TEMPLATE 構文は，テンプレートを定義します。

TEMPLATE 構文の有効域には，参照結合 (use association)，親子結合 (host association) 及びテンプレート引数結合 (template argument association) が適用されます[3]。

TEMPLATE 構文は，IMPLICIT NONE(TYPE, EXTERNAL) と書かれているかのように振る舞います (3.4.1)。つまり，TEMPLATE 構文内では，全ての型を持つ名前には明示的な型指定が必要で，全ての仮手続と外部手続には明示的引用仕様または EXTERNAL 属性が必要です。

TEMPLATE 構文は，（初期値設定プログラム単位を除く）プログラム単位の宣言部に書くことができます。TEMPLATE 構文は，宣言部とサブプログラム部から成ります。

TEMPLATE 構文の宣言部に書ける構文は，モジュールの宣言部に書ける構文と似ていますが，次のような点が異なります (3.4.1)。

- TEMPLATE 構文中に書けるもの
  - deferred 定数宣言文，deferred 手続宣言文，deferred 型宣言文 (3.1.1)
  - REQUIRES 文 (3.3)
  - REQUIREMENT 構文 (3.2)
  - TEMPLATE 構文（つまり，入れ子が許される）
  - INSTANTIATE 文 (3.5)

---

[3] テンプレート引数結合は，TEMPLATE 構文と INSTANTIATE 文との間の引数についての結合ルールですが，提案書 23-155r2 ではドキュメント化されていません。例えば INSTANTIATE 文の引数に “real(4)” と書いたとき，仮引数が deferred 定数なら組込み関数の引用（値は 4.0）と解釈され，仮引数が deferred 型なら「種別パラメタ 4 をもつ実数型」を表現すると解釈されます。

- TEMPLATE 構文中に書けないもの
  - IMPORT 文，IMPLICIT 文
  - DATA 文，FORMAT 文，ENTRY 文
  - ALLOCATABLE 文，ASYNCHRONOUS 文，BIND 文，CODIMENSION 文，CONTIGUOUS 文，EXTERNAL 文，NAMELIST 文，POINTER 文，PROTECTED 文，SAVE 文，TARGET 文，VOLATILE 文，VALUE 文
  - COMMON 文，EQUIVALENCE 文（廃止予定）
  - 変数の宣言

サブプログラム部は，CONTAINS 文と，それに続くテンプレートサブプログラムの並びです (3.4.2)。モジュール手続と同じように，テンプレート手続は，宣言部の言語要素や兄弟手続を親子結合によって参照します。

サブプログラム部には，EQUIVALENCE 文または COMMOM 文を書くことはできません (3.4.2)。

TEMPLATE 構文は，組込み手続，明示的引用仕様をもつ手続，及び演算子だけを参照します (3.4.3)[4]。

3.5 INSTANTIATE 文

INSTANTIATE 文は，テンプレートの具体化を指定します。また，テンプレートの deferred 引数に結合する具体化引数 (instantiation argument) を指定します。引数はキーワード引数になることができます。

INSTANTIATE 文により，テンプレート内が全て具体化され，テンプレート内の PUBLIC 属性をもつ名前，定義演算子と定義代入及び定義入出力が参照できるようになります。ONLY がないとき，それらの全てが参照でき，ONLY があるとき，それらのうち指定したものだけが参照できます。また，名前や定義演算子に局所名（別名）を指定するとき，その局所名で参照します。

4. その他の変更点

Fortran 2023 を次のように変更します。ここで，R の付いた番号と箇条番号は，Fortran 2023 国際規格のものです。

- R508 宣言構文 (specification-construct) に，TEMPLATE 構文と INSTANTIATE 文を追加
- R702 型指定子 (type-spec) に，deferred 型 (deferred-type) を追加
- R703 宣言型指定子 (declaration-type-spec) に，"TYPE (deferred 型)" を追加

---

[4] 原文の直訳です。正確な意味を確認する必要があります。

- R728 型属性指定子 (type-attr-spec) の "EXTENDS (親型名)" において，親型名は deferred 型であってはならない。
- 全体に，次元数引継ぎ指定 (assumed-rank-spec) を，assumed-or-deferred-rank-spec に変更。deferred-rank 言語要素は deferred 定数であるとする。
- R829 rank-clause の制限事項 C864 を緩和：rank-clause で宣言される言語要素として名前付き定数を許す。
- deferred 引用仕様本体は，IMPORT 文が書いてあるかのように，親有効域中の言語要素を参照可能。
- "10.1.12 定数式 (constant expression)" において，定数式として deferred 定数を許す。
- 引用仕様宣言 (interface block) の引用仕様部構文 (interface-specification) に，deferred 引用仕様本体 (deferred-interface-body) を許す[5]。
- R1522 手続特定子 (procedure-designator) に，deferred 手続 (deferred-proc) を追加
- R1524 実引数 (actual-arg) に，deferred 手続 (deferred-proc) を追加

[5]引用仕様宣言中に引用仕様本体と deferred 引用仕様本体との両方を許すことになります。これらは見かけ上区別がつかない場合があるので、構文解析で問題がないか確認する必要があると思われます。