

第2章 記述法と構文

本章では、本書で使われている記述規約と HPF 指示文の構文について記述する。

2.1 記述法

本書では Fortran 95 規格と同じ記述法を使用する。特に、構文規則には同じ規約を使用する。言語機能の BNF 記述は Fortran 規格と同じ様式で与えられる。HPF 構文規則と Fortran 構文規則を区別するために、各 HPF 規則は $Hsnn$ という形式の識別番号をもつ。ここで、 s は章番号に対応しており、 nn は 2 桁の順序番号である。本書で定義されてない非終端記号は Fortran 規格の中で定義されている。また、「記憶単位」のようないくつかの技術用語は Fortran 規格で定義されていることにも注意されたい。

第 1 章で記したように、文中の F95:2.4.7 の形式で示される参照は、Fortran 95 規格の 2.4.7 への参照を表している。

第 III 部は公認拡張仕様を記述する。ここでは以前の章ですでに導入されている構文規則を拡張しなければならない場合がある。特に、ここでの構文規則はしばしば第 II 部の類似の構文規則の拡張されたものである。このような場合には、非終端記号の名前は接尾辞 (suffix) *-extended* がついている。したがって、*name* のような非終端記号が再定義されたとき、構文規則の残りの中で、*name* のすべての引用は *name-extended* で置き換えられるという条件の下で、それは *name-extended* として引用される。

第 II 部の制約などが公認拡張仕様によって修正される場合には、文章中にそのように記述され、前方への参照が与えられる。そのような前方参照が目立つように、余白に (ここに示すように) 下向き矢印が表示される。

(第 III 部の) 各修正は、修正される第 II 部の元の言語定義への後方参照を含む。このような後方参照が目立つように、余白に (ここに示すように) 上向き矢印が表示される。

【仕様の根拠】 本書の全体を通して、機能を含むこと、特定の機能定義を選択したこと、その他の決定を行ったことの根拠を説明する資料は、この形式で仕切られる。言語定義にのみ興味がある読者はこの部分を読み飛ばすことを望むかもしれない。一方、言語の設計に興味がある読者はこの部分をより注意深く読みたいかもしれない。【以上】

【利用者への助言】 本書の全体を通して、主に利用者のための資料 (構文と解釈の多くの例を含む) はこの形式で仕切られる。技術的な資料にのみ興味がある読者はこの部分を読み飛ばすことを望むかもしれない。一方、より教育的な方法を求める読者はこの部分をより注意深く読みたいかもしれない。【以上】

【実装者への助言】 本書の全体を通して、主に実装者のための資料はこの形式で仕切られる。言語定義にのみ興味がある読者はこの部分を読み飛ばすことを望むかもしれな

い。一方、コンパイラの実装に興味がある読者はこの部分をより注意深く読みたいかもしれない。【以上】

2.2 指示文の構文

HPF 指示文は次の意味で Fortran 構文と一貫性がある。すなわち、ある HPF 指示文が将来の Fortran 規格の一部として採用された場合、HPF プログラムを変換するために必要な変更は指示文先頭語 (*directive-origin*) を空白に置き換えるだけである。

H201 *hpf-directive-line* is *directive-origin hpf-directive*

H202 *directive-origin* is !HPF\$
or CHPF\$
or *HPF\$

H203 *hpf-directive* is *specification-directive*
or *executable-directive*

H204 *specification-directive* is *processors-directive*
or *align-directive*
or *distribute-directive*
or *inherit-directive*
or *template-directive*
or *combined-directive*
or *sequence-directive*

H205 *executable-directive* is *independent-directive*

制約: *hpf-directive-line* には、他の文があってはならない。

制約: *specification-directive* は、宣言構文 (*declaration-construct*) が現れてもよい場所のみ現れてもよい。

制約: *executable-directive* は、実行構文 (*executable-construct*) が現れてもよい場所のみ現れてもよい。

制約: *hpf-directive-line* は、プログラム単位内の周囲の形式に従って、Fortran の自由形式 (F95:3.3.1.1) または固定形式 (F95:3.3.2.1) の注釈行の規則のどちらかに従う。(F95:3.3)

hpf-directive は、文字種 (英大文字、英小文字) を区別せず、自由形式の空白の規則 (3.3.1) に従う。これは、固定形式の HPF プログラムの中でもそうである。しかし、HPF 規格合致の言語処理系は、HPF 指示文の中にある余分なまたは足りない空白を見つけ出す必要はない。Fortran の規則によって、自由形式の *directive-origin* は !HPF\$ でなければならないことに注意されたい。HPF 指示文は継続してよい。その場合、各継続行もまた *directive-origin* で始まらなければならない。継続する HPF 指示行の間に他の文を置いてはならない。HPF 指示行は継続行に現れてはならない。HPF 指示行は後に注釈があってもよい。

1 どちらのプログラム形式でも、キーワード END FORALL および NO SEQUENCE の中にあ
2 る空白は任意である。

3 自由形式の HPF 指示文の継続の例:

4
5 !HPF\$ ALIGN ANTIDISESTABLISHMENTARIANISM(I,J,K) &
6 !HPF\$ WITH ORNITHORHYNCHUS_ANATINUS(J,K,I)

7
8 以下に、固定形式の HPF 指示文の継続の例を示す。継続を意味する場合以外、第 6 けた
9 は空白でなければならないことに注意する。

10
11 !HPF\$ ALIGN ANTIDISESTABLISHMENTARIANISM(I,J,K)
12 !HPF\$*WITH ORNITHORHYNCHUS_ANATINUS(J,K,I)

13
14 次の例は、自由形式としても固定形式としても扱える「汎用的」な HPF 指示文の継続方
15 法を示している。最初の行の “&” は第 73 けたにあることに注意されたい。

16
17 !HPF\$ ALIGN ANTIDISESTABLISHMENTARIANISM(I,J,K) &
18 !HPF\$&WITH ORNITHORHYNCHUS_ANATINUS(J,K,I)

19
20 第 III 部では、HPF 2.0 の公認拡張仕様として、新しい宣言指示文と実行指示文を導入
21 する。以下にこれらを記す。

22
23
24 H206 *specification-directive-extended* **is** *processors-directive*
25 **or** *subset-directive*
26 **or** *align-directive*
27 **or** *distribute-directive*
28 **or** *inherit-directive*
29 **or** *template-directive*
30 **or** *combined-directive*
31 **or** *sequence-directive*
32 **or** *dynamic-directive*
33 **or** *range-directive*
34 **or** *shadow-directive*

35
36
37
38 H207 *executable-directive-extended* **is** *independent-directive*
39 **or** *realign-directive*
40 **or** *redistribute-directive*
41 **or** *on-directive*
42 **or** *resident-directive*
43

44 以下の規則は、Fortran 95 の規則 R215 の拡張である。
45
46
47
48

H208	<i>executable-construct-extended</i>	is	<i>action-stmt</i>	1
		or	<i>case-construct</i>	2
		or	<i>do-construct</i>	3
		or	<i>if-construct</i>	4
		or	<i>where-construct</i>	5
		or	<i>on-construct</i>	6
		or	<i>resident-construct</i>	7
		or	<i>task-region-construct</i>	8
				9
				10
				11
				12
				13
				14
				15
				16
				17
				18
				19
				20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				32
				33
				34
				35
				36
				37
				38
				39
				40
				41
				42
				43
				44
				45
				46
				47
				48