4011：504（コーディングルール；
プログラミング書法；Fortran；数値予報）

標準コーディングルール*

室井ちあし[*1]・豊田英司[*2]・吉村裕正[*3]
保坂征宏[*4]・杉正人[*5]

1．はじめに

一般に，多機能で使いやすいプログラムを作成しようとすると，そのソースコードは混沌としたものになってしまう．限られた目的に利用されるプログラムならば動けばそれでよいという場合もあるが，長年にわたりプログラムを利用・発展させ，コミュニティーの知見の集大成にしたいという場合は，それにふさわしい書法規範を策定する必要がある．

著者らがその研究開発に携わっている数値予報モデルにおいても，数値天気予報が開始されて以来，モデルは年とともに着実に複雑になってきており，いわばこれまでの技術開発の集大成である．しかも，一見さんはお断りとばかり，ベクトル化や並列化等の計算技術に追随するための様々な変更も加えられ，そのソースコードを眺めて理解することすら容易ではなくなってきている．そして複雑化の傾向は今後もとどまるところを知らず，さらなる高分解能化への対応や，他のモデルとの結合やネスティング，物理過程のモデルの開発・改良，そしてモデルを用いたアプリケーションといった開発課題が山積している．一方で数値予報モデルを取り巻く環境も急速に変化しており，気象庁モデルの大学・研究機関等への公開も開始されていることから，複数の機関に分散した研究者が共同して同じソースコードを用いた開発を行い，知見を蓄積しようとする機会が今後ますます増加すると予想される．

そこで，ソースコードの解読やバグ出しを助け，その維持管理，改変や交換を容易にする方策を考えておく必要がある．ソースコードの書法の指針，すなわちコーディングルールが策定されるべきだと考えられる．

我々はFortranで書かれたプログラムに囲まれて暮らしている．しかしながら，現代の計算機科学者はFortranをあまり使わない（本屋でもFortranに関する書籍は数少ない）ので，今後もFortranを使うつもりならば，気象の数値計算に携わる我々自身が経験を蓄積し，望ましい規範を定めていくしかない．

実際に英国気象局ではコーディングルール“European Standards For Writing and Documenting Exchangeable Fortran 90 Code”（http：//www.metoffice.com/research/nwp/numerical/fortran90/f90_standards.html）が策定されているほか，オクラホマ大学のメソスケールモデルARPSでは“New ARPS Coding Standards with Fortran 90”（http：//caps.ou.edu/ARPS/coding/），そして米国で共同開発が進められているWRFでもFortran90の採用に伴い独自の規則“WRF Coding Conventions”（http：//www.mmm.ucar.edu/wrf/WG2/WRF_conventions.html）がそれぞれ定められている．

日本においても，1999年ごろから気象庁予報部数値予報課と気象研究所気候研究部・予報研究部の有志が中心となって標準コーディングルールを作成する活動を行っている．ルールの策定にあたっては，数値予報で使われる予報モデルやデータ同化のプログラムへの適応を想定はしているが，高速で実行する必要がある大規模なソフトウェアを多人数で作成する場合には，

* Standard coding rule.

*1 Chiashi MUROI，気象研究所予報研究部．
*2 Eizi TOYODA，気象庁予報部数値予報課．
*3 Hiromasa YOSHIMURA，気象研究所気候研究部．
*4 Masahiro HOSAKA，気象研究所気候研究部．
*5 Masato SUGI，気象研究所気候研究部．

異分野でも役に立つと考えている．そしてこの成果をhttp：//www.mri-jma.go.jp/Dep/fo/mrinpd/coderule.html で公開している．以下，2001年10月現在の内容についていくつかのトピックスを紹介する．細部や実例等については割愛する．また最新状況については随時 Web をご覧いただきたい．

2．スタイルルール（推奨リスト）

ソースコードを読みやすく，またバグ発見を助けるために，いくつかの推奨するスタイルルールを制定している．

○フリーフォーマット（自由形式）

7-72桁に文を書かなければならない等の制約がなくなり，それ以外の桁位置に書いた文字が無視されるなどの誤りを防ぐことができる．

○１ファイルには同名のプログラム単位１つを格納

プログラム単位とは，主プログラム，外部関数，外部サブルーチン，モジュール，BLOCK DATA のいずれかである．たとえば，外部サブルーチン SUB は sub.f90に保存されるべきである．こうしておけばプログラムを読むときにファイルを探しやすい．また，サブルーチンを書き換えた場合に他のプログラム単位をコンパイルしなおすことがなくなるので，コンパイル時間が短くなる．

○名前

いつも主プログラムは program 文ではじめること．また end 文としては“end program name”や“end subroutine name”等の形を使用すること．

○継続記号

継続行を追加する場合の継続記号（&：アンド記号）は，行末（開始行の終わりの文字として）だけではなく行頭（継続行の最初の文字として）にもつけること．継続行であるということがはっきりと区別できるようになる．

○字下げ（インデンテーション）

読みやすさのため，２桁づつの幅で字下げすること．字下げとはプログラムの構造上ひとまとまりとなる文をそのほかの文に比べて２字右側の桁位置から書き始めることである．字下げを行う「まとまり」とはプログラム単位（subroutine），引用使用宣言（interface），実行部ブロック（do, if など）である．これらの終わりの文はすべて「end」で始まる文なので，「end まで字下げ」と覚えれば簡単である．字下げされた範囲の中の注釈行も字下げすること．これによって，行頭の空白でない最初の桁位置を追うだけでプログラムのアルゴリズム構造が見出せるようになる．継続行は開始行に対して２字字下げすること．

○空白

読みやすさのため，名前と演算子など，字句の間に適切な空白を入れ，同じスタイルを一貫して用いること．英文タイプの句読法に従うのが望ましい．またブロックではない一群の文は前後の文との間に空行を入れて区別すること．

空白は原則として１個とするが，似たような構成の代入文が連続するときに余分な空白をいれて等号の桁位置をそろえ，類似性を見比べやすくするなどしてもよい．

○implicit none

すべてのプログラム単位の冒頭（use 文があるときはその直後）に implicit none 文を書くこと．そしてすべての変数・定数は明示的に宣言すること．これによって，宣言されていない名前を用いるとコンパイルエラーを起こすので，名前のつづり間違えによるバグを防ぐことができる．

○１行に複数の文の禁止

１行に複数の文を書かないこと．プログラムがわかりにくくなる．

○意味がわかる名称

変数名やサブルーチン名は，その役割に基づいた名前をつけておくとプログラムを読む際にその意味や構造を把握しやすい．

○変数の宣言方法

FORTRAN 77では型宣言文と属性を与えるための文（SAVE, DATA, DIMENSION, PARAMETER, など）が別であったため，同じ名前の列を何度も書く必要があった．Fortran 90では型宣言文で属性をすべて設定できるようになったため，すべての属性を型宣言文で指定すること．

○引数の入出力特性

識別を容易にするため intent（in），intent（out），intent（inout）をつける．例えば intent（in）がつけられた変数の値を変更する文が存在すると，コンパイル時にエラーが発生し，デバッグを助ける．なお，手続のモジュール化または interface ブロックによって引用仕様を明示することにより，さらにその恩恵を受けることが可能である．

○入出力のフォーマット情報

文番号を伴った FORMAT 文は禁止する．代わりに

入出力文の fmt 指定子に記述すること.

○配列表記

プログラムの行数が少なくなるので，可能であれば使用すること．配列か変数かの区別をつけて読みやすくするため，カッコ内に配列の形状を示すこと．並列最適化コンパイラの動作を助け，バグを少なくするため，配列代入文の上下限指定は省略しないほうがよい.

○論理比較

従来の.GT., .GE., .EQ., .LT., .LE., .NE. の代わりにそれぞれ>, >=, ==, <, <=, /=を使用する．数学表記に近いこの新しい構文の方が意味がわかりやすい.

○GOTO 文

GOTO 文を使うとプログラムがわかりにくくなることが多い. Fortran に限らず，ほとんどのプログラムはループと条件分岐だけで書くことができ，そうすることが望ましいと考えられている（構造化プログラミング）. FORTRAN 77では固定長のループしか表現できないため GOTO 文を用いる必要があったが，do のあとの繰り返し指定を省略すると無限ループを表し，ループの途中からの脱出は exit 文で行えるようになったので，構造化プログラミング理論で用いられるすべてのループが GOTO 文なしで書ける.

3．Fortran90の機能を生かして

3.1 モジュールの使い方

モジュールは Fortran90に新しく付け加えられた機能のうちもっとも重要なものの１つである．モジュールを使うことによって，意味的にひとまとまりである１つあるいはそれ以上のプログラムが COMMON よりもスマートな形でデータを共有し，外部から独立した形で存在することが可能となる．しかし，モジュールは多機能であるがゆえに，乱用されると他人にはわかりにくいプログラムになるであろう．著者らは一定の指針が必要であると考える.

以下では著者らが考えるルールを記すが，より一般的なプログラムについて必ずしも適当かどうかはわからない．知見を集積しつつ検討している段階であることに留意して欲しい.

3.1.1 モジュールについての基本的なルール

○private/public 属性

全てのモジュールの冒頭に private を書くこと．これにより全てのモジュール変数やサブルーチンはデフォルトでは外部から参照できなくなる．外部に公開することが必要なもののみに public 属性を付けること.

○only 句

モジュールを参照する場合，単に“use モジュール名”と書くのではなく，only 句をつけて必要なもののみを参照可能にすること.

○モジュール変数の値の設定・変更方法

モジュール変数の値の設定・変更は，モジュール内部のサブルーチンにより行うこと．必要なデータは，外部から引数を通じて得てもよいし，内部でファイルから読み込んでも構わない.

外部のプログラムからモジュール変数を use 文で参照することは可能であるが，そこで値の変更をすることを禁じる．これは，モジュール変数がどこでどういうタイミングで変更されるか分かりにくくなることを防ぐためである.

○総モジュール化

全てのサブルーチンはモジュールに属することとする．（場合によっては，サブルーチンの前後に module 文と end module 文がつくだけかもしれない.）これにより，コンパイル時にコンパイラが引数の型のチェックを行いバグを発見するのに役立つことが期待される.

○モジュールの階層作りと参照関係の規則

総モジュール化を実現した場合，各モジュールのコンパイル時に，use 文で参照する全てのモジュールがコンパイル済みであることが必要になる．複数のモジュール間の参照が野放図に行われると，言語仕様では禁止されている循環的参照が生じる危険性がある．プログラムの構成にあたっては，将来の改変の際に困ることのないような設計が必要である.

分かりやすい方法として，メインプログラムを最上位，他のどのサブルーチンも呼び出さないユーティリティー的なプログラムを最下位とするような階層構造をもつ構成にし，上位の階層のモジュールが下位の階層のモジュールを参照することは許すが逆は許さない，といった規則を定めることが考えられる.

3.1.2 モジュールの３つの類型

モジュールの使用方法を以下の３つの類型に限定する．(1), (2)はパッケージ全体から参照可能なモジュール変数を提供するための，相対的に下位のモジュールである．(3) は意味的にひとまとまりのパッケージの外部へのインターフェースを提供するためのものであり，相対的に上位のモジュールであることが多い.

(1) 定数参照型モジュール

定数参照型モジュールでは，複数のルーチンで使用する定数（parameter 属性を持つ）を宣言し，値を設定する．定数としては例えば，物理定数があげられる．これらの定数の値はコンパイル時に読み込まれるので，最適化が行われやすいと期待できる．

(2) 変数参照型モジュール

ジョブによって値が異なる等の理由で定数にはできない，ジョブ内部では一定の値をもつような（実行ごとに違ってもよい）モジュール変数を複数のルーチンに提供するためのものである．

変数参照型モジュールは，モジュール変数と，それらの値を設定するための値設定用サブルーチンからなる．プログラム実行時のはじめに一度値設定用サブルーチンが呼ばれ，モジュール変数の値が設定される．モジュール変数は，外部のいろいろなサブルーチンから use 文によって参照可能である．

実行中にモジュール変数の値を変更する必要がある場合は，この変数参照型モジュールではなく，(3)パッケージ型モジュールを使用する．

(3) パッケージ型モジュール

初期化サブルーチン・実行サブルーチン・局所サブルーチン・モジュール変数からなる．

初期化サブルーチンでは，モジュール変数の初期値の設定を行う．実行サブルーチンはパッケージ型モジュール内に1個もしくは複数個あり，引数を通じて外部からデータを入力し計算を行い，引数を通じて外部に必要なデータを出力する．また，実行サブルーチンではモジュール変数の値の変更を行ってもよい．この初期化・実行サブルーチンのみに public 属性をつけて外部に公開する．モジュール変数と，モジュール内部でのみ呼び出される局所サブルーチンには private 属性をつける．モジュール変数を外部から参照する際は実行サブルーチンの引数を通じて行う．

このように外部に公開する部分を制限しパッケージの独立性を高めることにより，プログラム全体の構造が分かりやすくなり，多人数で大規模なプログラムの共同開発を行うことが容易になると期待される．

3.2 動的割り付け

サブルーチン内のワーク的変数を動的にメモリに割り付けることにより，プログラム作成者はワーク的変数の管理から開放され，メモリを有効利用することができる．また実行時に配列の大きさを変更できるので，解像度に依存しない数値モデルが作成できる．

動的割り付けを行うには，自動割り付け配列を用いる方法と，allocatable 属性をつけて宣言し allocate 文で割り付けを行う方法とがある．サブルーチン内でワーク的に割り付ける場合は前者で，恒久的に割り付ける場合は後者で行うこととする．

動的割り付け機能を用いると，主記憶の割り付け・解放の動作を行う等のために静的割り付けの場合よりも実行時間が若干長くかかることがわかっている．割り付けの際に主記憶がどのように使用されるかは処理系依存と思われるので，一般的な指針を定めるためにはより多くの環境についての調査や知見の蓄積が必要である．

4．避けるべき機能

これまで述べたように，望ましいスタイルのルールを提案する一方，時代遅れの機能や廃止予定事項など，避けるべきものを規定している．

時代遅れの機能とは EQUIVALENCE, COMMON, BLOCKDATA, ENTRY, INCLUDE, DO WHILE, 文関数，組み込み手続きの個別名などである．

また既に Fortran 90で廃止予定事項とされた機能も避けるべきである．これには算術 IF 文，不整構造 DO 構文，割り当て型 GOTO 文および ASSIGN 文，H 型編集記述子，end if 文への分岐，選択戻り，PAUSE 文が該当する．

Fortran 規格にはないタブ文字の使用や，処理系固有の拡張構文については可搬性を損なうので，避けるべきである．

5．おわりに

ここまで，コーディングルールに定めている規則について解説してきた．現段階では規定していないが，プログラム構築の際には重要な事柄で本ルールに記載すべきである事項も依然多くあると認識している．それらの事項については，今後さらに経験を積み重ねることにより，順次追加していきたいと考えている．

ルールを明瞭にするために守るべき事柄を条文化し，序列を整理，詳細な説明や用語解説を別立てにして拡充することも検討している．

現時点においては，コーディングルールの必要性についてなるべく広範な理解を得て，実現が容易なものから実際のソースコードの記述に共通ルールを適用していくことがまず必要であると考えている．その上で必要に応じてルールを再検討していくなど，ルールの

修正や追加（あるいは削除）についても柔軟に対応していきたい．読者からのコメントや意見，情報等も歓迎する．

謝　辞

本ルール構築に向けた取り組みを行うにあたっては，科学技術振興調整費「高精度の地球変動予測のための並列ソフトウェア開発に関する研究」に関わる様々な活動において，各方面から多くの助言・激励をいただいた．関心を寄せていただいているすべてのみなさまに感謝する．

新刊図書案内

表　題	編　著　者	出　版　者	出版年月	定　価	ISBN	備　　考
学会へ行こう！ ：市民のための理科系学会参加マニュアル	山下美智子	文葉社	2001.11	¥1,200	4-9980907-3-9	
環境流体シミュレーション	河村哲也 菅　牧子 桑原邦郎 小紫誠子	朝倉書店	2001.11	¥4,700	4-254-18009-8	CD-ROM付き
気象データひまわり CD-ROM2002	日本気象協会	丸善	2001.11	¥12,000	4-621-04931-3	
気象予報士試験精選問題集：平成14年版	気象予報士試験研究会	成山堂書店	2001.11	¥2,800	4-425-97367-4	
小学生の「地球環境」大疑問100： 目からうろこ	環境goo	講談社	2001.11	¥1,500	4-06-210817-8	
雨の事典： 空と海と大地をつなぐ	レインドロップス	北斗出版	2001.12	¥2,500	4-89474-021-4	
新・天気予報の手引	安斎政雄 日本気象協会	クライム気象図書販売部	2001.12	¥1,515	4-907664-38-9	改訂30新版
地球・気象		学研	2001.12	¥2,000	4-05-500422-2	ニューワイド学研の図鑑
雪と氷の自然観察	日本自然保護協会	平凡社	2001.12	¥2,000	4-582-54017-1	フィールドガイドシリーズ7

注：表中で定価はすべて本体価格です（特記したものを除く）．