
 報 告

日本工業規格案

電子計算機プログラム用言語 COBOL について*

COBOL/JIS 原案作成委員会

1. はじめに

電子計算機プログラム用言語 COBOL の日本工業規格案（以下 JIS COBOL 原案と略称する）が、本学会の COBOL/JIS 原案作成委員会において作成されたので、その制定の経過や内容の概要について簡単に報告する。

2. 制定の経過

電子計算機プログラム用言語の JIS 規格としては、ALGOL と FORTRAN がすでに制定され、その役割が十分に果たされた。これはすでに3年以上経過したので、内容の再確認の作業が行なわれている。事務用共通言語として広く使われ、また作られている COBOL の規格化も以前から要望されていた。

1969 年春に本学会は、工業技術院から JIS COBOL 原案作成の委託を受けたが、その年は種々の事情から原案作成の作業は見送られた。翌 1970 年 2 月になり COBOL/JIS 原案作成委員会が本学会の規格委員会の下に設置され〔主査：水野幸男（日電）〕、具体的な原案作成作業が開始された。

COBOL/JIS 原案作成委員会のおもな委員は、つぎのような顔ぶれだった。

水野 幸男（日電、主査）
 伊藤 公一（東芝）
 伊藤 靖彦（日電、幹事）
 小碓 暉雄（三菱）
 城戸 健（日本放送協会）
 黒河亀千代（工技院）
 徳永 英二（日本アイビーエム）
 西村 恕彦（電総研）
 古本 俊爾（沖電気）
 細谷 僚一（通研）
 松本 恵一（日立）
 毛利 良男（富士通）
 山谷 得嗣（日本ソフト）

以下は、種々の事情によって、作業の一部分にだけ

参加した人達である。

植村 俊亮（電総研）
 大駒 誠一（慶大）
 近藤 久（通研）
 竹本 宏和（国鉄）
 芳賀 敏郎（山陽バルブ）
 広川 陽一（鹿児島データセンター）
 山田 勲（日本ユニバック）
 吉村鉄太郎（管理工学研）

委員会は 1971 年 3 月までの約 1 年間に、本会議と副会議を約 30 回開催し検討を重ね、1971 年 3 月に原案本文の作成を完了した。

COBOL の言語仕様は、CODASYL のプログラミング委員会において、その開発・保守作業が続けられている。また、その標準化に関しては、アメリカ規格協会 (ANSI) が中心となり、1967 年 1 月現在の CODASYL COBOL にもとづいて、国際推薦規格案 (ISO DR 1989 COBOL)** が制定されている。

このような状況において、COBOL/JIS 原案作成委員会は、JIS COBOL 原案を作成するにあたり、つぎのような基本方針を立てた（なお、その後にとった態度についても同時に述べる）。

(1) JIS COBOL 原案の技術的内容は国際推薦規格案に基づいて作成する。——1967 年 1 月以降に CODASYL が行なった COBOL の改訂部分は、規格の一貫性の観点から、JIS COBOL 原案には含めないことにした。

(2) 国際推薦規格案が現在の CODASYL COBOL と違う点、および仕様があいまいな点は、解説や参考などで指摘する。——現在の CODASYL COBOL で削除されている機能や、国際推薦規格案であまりになっていない仕様は、できるだけ指摘した〔ただし、(1) の方針はくずさなかった〕。

(3) 小型 COBOL に関する検討を別途行なう。

* On Proposed Japanese Industrial Standard COBOL

** ANSI: COBOL, X 3.23-1968, 日本規格協会抜いてみるとその内容がわかる。

— 国際推薦規格案の最低水準が日本の小型 COBOL の現状に合うかどうかを検討した (特に, COMPUTE 命令を低水準に入れるべきかが問題となった). その結果, 小型 COBOL の規格化は進めるべきであるが, 当面の JIS COBOL 原案作成の作業においては, あとまわしとすることにした.

(4) 用語の訳語の統一をはかる. — 用語の訳語を統一する場合の主眼点は, できるだけ日本語化することであった. さいわい COBOL 1965 年版の翻訳が本学会から刊行されていたので, これをもとにほかの JIS 規格などを参考にしながら訳語を統一した.

3. JIS COBOL 原案の概要

JIS COBOL 原案は A 4 版 21 章 272 ページにわたるため, ここにその全体をかかげることはできない. したがってここでは, 各機能単位の概要を, 水準間の違いをも含めて示す. 以下は, JIS COBOL 原案のおもな章の序説にある部分を抜粋したものである.

3.1 言語の構造

プログラムには, 見出し部, 環境部, データ部および手続き部を書く. それぞれの部を構成する言語要素

上記の 8 個の単位のうち, 中核以外のものを, それぞれ機能単位 (functional processing module) と呼ぶ.

(1) 中核の水準 中核は, 二つの水準 (level) に分ける. 低い水準は, 内部処理で基本的に必要な言語要素を含むものとし, 高い水準に対する部分集合とする.

(2) 機能単位の水準 機能単位は, それぞれ二つまたは三つの水準に分ける. どの機能単位においても, 低いほうの水準は, 高いほうの水準の部分集合とする. ある機能単位においては, 最低の水準は, 空集合とする.

(3) 水準の組合せ方 中核および機能単位は, データ処理応用の広い範囲の要求を満足させるために, それぞれ一つずつの水準を選んで, 自由な組合せを作って利用してよい.

(4) COBOL の作成 ある COBOL の言語仕様を定め, それにもとづいて処理系を作り, 利用者に提供する者を, 作成者 (implementor) と呼ぶ.

COBOL を作成するとき, それが, 中核と機能単位との各水準を, この規格で規定されたように含ん

COBOL の規格の構造図

| 中核 | 機能単位 | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|
| | 表操作 | 順呼出し | 乱呼出し | 分類 | 報告書作成 | 区分化 | 登録集 |
| 2 中核 1,2 | 3 表操 1,3 | 2 順呼 1,2 | 2 乱呼 0,2 | 2 分類 0,2 | 2 報告 0,2 | 2 区分 0,2 | 2 登録 0,2 |
| 6 章 | 9 章 | 11 章 | 13 章 | 15 章 | 17 章 | 19 章 | 21 章 |
| 1 中核 1,2 | 2 表操 1,3 | 1 順呼 1,2 | 1 乱呼 0,2 | 1 分類 0,2 | 1 報告 0,2 | 1 区分 0,2 | 1 登録 0,2 |
| 8 章 | 10 章 | | 12 章 | 14 章 | 16 章 | 18 章 | 20 章 |
| 5 章 | 1 表操 1,3 | | 空 | 空 | 空 | 空 | 空 |
| | 7 章 | | | | | | |

(language element) は, つぎに示す 8 個の単位のうちのひとつ以上に属するものとする.

- 中核 (nucleus)
- 表操作 (table handling)
- 順呼出し (sequential access)
- 乱呼出し (random access)
- 分類 (sort)
- 報告書作成 (report writer)
- 区分化 (segmentation)
- 登録集 (library)

ていれば, その COBOL は, ここに定める COBOL の作成上の要求を満たしているものとみなす.

この規格で定める COBOL の最低水準および最高水準は, つぎのとおりとする.

(a) 最低水準 (minimum JIS COBOL) は, 中核と機能単位とのそれぞれの最低の水準 (注 1) を組み合わせたものとする.

(b) 最高水準 (maximum JIS COBOL) は, 中核

(注 1) 空集合の水準を除けば, 中核, 表操作, 順呼出しの低い水準からなる.

と機能単位とのそれぞれの最高水準を組み合わせたものとする。

3.2 中核

中核 (nucleus) は、プログラムの4個の部における基本構造のわく内で、データを処理するための基本的な言語機能を備える。

(1) **水準1の特性** 中核の水準1 (nucleus level 1) は、修飾、句読文字、データ名の書き方、連結語および表意定数について、制限された機能を有する。中核の水準1の手続き部は、ACCEPT, ADD, ALTER, DIVIDE, DISPLAY, IF, MOVE, PERFORM および SUBTRACT 命令では制限つきの機能、ENTER, EXAMINE, EXIT, GO, MULTIPLY, NOTE および STOP 命令では完全な機能を有する。

機能単位と組み合わせられた中核の水準1は、各機能単位の制限にしたがう。中核の水準1と組み合わせられた機能単位は、中核の水準1の制限にしたがって、その機能のすべてを利用できる。

(a) **形式の記法** この水準は、分離符セミコロン、句読文字コンマおよび句読文字セミコロンを含まない。

(b) **名前の特性** すべてのデータ名は、英字で始まらなければならない。修飾は、使えない。すべての名前は、一意でなければならない。

(c) **表意定数** 使える表意定数は、つぎのとおりとする。

ZERO
SPACE
HIGH-VALUE
LOW-VALUE
QUOTE

(d) **正書法** 語および数字直定数は、2行にまたがって書いてはならない。

(2) **水準2の特性** 中核の水準2 (nucleus level 2) は、修飾、句読文字、データ名の書き方、連結語および表意定数について、すべての機能を有する。

機能単位と組み合わせられた中核の水準2は、各機能単位の制限にしたがう。中核の水準2と組み合わせられた機能単位は、その機能のすべてを利用できる。

3.3 表操作

表操作機能単位 (table handling module) は、一連のデータ項目の表を定義し、表のなかの相対的位置によって項目を呼び出す。項目は、何回繰り返されるかを指定できる。各項目は、添字か指標名を用いて一意

にして参照する。

(1) **水準1の特性** 表操作の水準1 (table handling level 1) は、1次元固定長の表を呼び出す機能を有する。

(2) **水準2の特性** 表操作の水準2 (table handling level 2) は、3次元固定長の表を呼び出す機能を有する。この水準は、複数の指標名で指標づけができる。複数の指標名または一意名に、同時に値が設定できる。指標名の内容を加算または減算によって変更できる。

(3) **水準3の特性** 表操作の水準3 (table handling level 3) は、3次元可変長の表を呼び出す機能を有する。また、この水準は、昇順あるいは降順のキーについて、指定された条件を満足する項目の表引きができる。

3.4 順呼出し

順呼出し機能単位 (sequential access module) は、ファイルからレコードをきまった順序で呼び出す。その順序は、ファイルにレコードを書き出したときにきまる。また、この機能単位は、再運転点 (rerun point) および複数のファイル間での記憶領域の共用を指定する。

(1) **水準1の特性** 順呼出しの水準1 (sequential access level 1) は、標準ラベルまたはラベルなし、再運転、複数装置ファイルの処理、入出力領域の共用、可変長レコード、大記憶装置の1個の区域に順呼出しファイルを割り付けることなどの機能を有する。

(2) **水準2の特性** 順呼出しの水準2 (sequential access level 2) は、利用者のきめるラベル、利用者のきめるラベル処理手続き、不定入力ファイル、複数ファイルテープの位置づけ、バッファリングのための追加の入出力領域、可変長のブロック、ファイルの開閉時の付加処理、レコードの読み書き時の付加処理、利用者のきめる誤り処理手続き、大記憶装置の複数の区域に順呼出しファイルを割り付けることなどの機能を有する。

3.5 乱呼出し

乱呼出し機能単位 (random access module) は、プログラマが与えたキーにもとづいて、大記憶ファイルのレコードをランダムに呼び出す。また、この機能単位は、再運転点および複数のファイル間での記憶領域の共用を指定する。

(1) **水準1の特性** 乱呼出しの水準1 (random access level 1) は、標準ラベルまたはラベルなし、再

運転, 入出力領域の共用, 可変長レコード, 大記憶装置の1個の区域に乱呼出しファイルを割り付けることなどの機能を有する。

(2) **水準2の特性** 乱呼出しの水準2 (random access level 2) は, 利用者のきめるラベル, 利用者のきめるラベル処理手続き, 可変長のブロック, ファイルの開閉時の付加処理, レコードの読み書き時の付加処理, 利用者のきめる誤り処理手続き, 大記憶装置の複数個の区域に乱呼出しファイルを割り付けることなどの機能を有する。

3.6 分類

分類機能単位 (sort module) は, 利用者が指定するキーにしたがって, ファイル内のレコードの順序をそろえる。利用者は, レコードの追加, 削除, 新設, 変更, 編集, その他の修正の処理を, 入力手続きまたは出力手続きによって行なうことができる。この処理は, 分類によってレコードが並べかえられる前, あと, またはその両方で行なうことができる。

(1) **水準1の特性** 分類の水準1 (sort level 1) は, 一つの COBOL プログラムの1回の実行によって, 1個のファイルを1回だけ分類する機能を有する。そのファイルを分類する前, あと, またはその両方で, そのファイル中のレコードに対する特別な処理を行なうこともできる。

(2) **水準2の特性** 分類の水準2 (sort level 2) は, 一つの COBOL プログラムの1回の実行によって, 1個以上のファイルを, 1回以上分類する機能を有する。

3.7 報告書作成

報告書作成機能単位 (report writer module) は, 報告書の作成に要する詳細な手続きを指定するかわりに報告書の物理的な形態を指定することによって報告書の作成を容易にする。

報告書の論理的な構成を指定するのに, 階層構造の概念を用いる。報告書は, いくつかの報告集団 (report group) に分割され, さらにこの報告集団は, 一連の項目に分割される。このような階層構造によって, ある報告集団への参照は, その階層のレベルをも暗黙のうちに参照する。

一つの報告集団は, 一つ以上の項目を含み実際の行としては1行以上にわたって作成される。

(1) **水準1の特性** 報告書作成の水準1 (report writer level 1) は, 報告書頭書き, 報告書脚書き, ページ頭書き, ページ脚書きおよび明細の各報告集団の

型を指定する機能を有する。さらに一つのファイルについてはただ一つの報告書を記録する機能を有する。

(2) **水準2の特性** 報告書作成の水準2 (report writer level 2) は, 制御集団の型, 自動的な集計と破算, 報告集団が作成表示される直前に実行される利用者がきめる手続き, 合計報告の作成, 集団表示, 並び指定などの機能を有する。さらに, 一つのファイルに対して一つ以上の報告書を記録する機能を有する。

3.8 区分化

区分化機能単位 (segmentation module) は, 実行用プログラムのオーバーレイを可能にする。

(1) **水準1の特性** 区分化の水準1 (segmentation level 1) は, 常駐区分 (permanent segment) と独立区分 (independent segment) を指定する機能を有する。同じ優先番号をもつすべての区分は, 原始プログラムのなかで隣接していなければならない。すべての常駐区分は, 原始プログラムのなかで隣接していなければならない。

(2) **水準2の特性** 区分化の水準2 (segmentation level 2) は, 異なる優先番号をもつ節を混在させる機能をもち, 原始プログラムの固定部分にオーバーレイ可能な区分を入れることができる。

3.9 登録集

登録集機能単位 (library module) は, 指定された原文 (text) を登録集から複写する。

COBOL 登録集は, 原始プログラムが翻訳されるときに利用できるプログラムの原文を含む。登録集原文を翻訳した結果は, その原文が原始プログラムの一部として書かれているのと同じとする。

COBOL 登録集は, 環境部 (environment division), データ部 (data division) および手続き部 (procedure division) のための原文を含み, COPY (複写) 命令によって引用できる。

COBOL 登録集原文は, COBOL プログラムとは独立な機能によって, 作成者の定める方法で COBOL 登録集に登録される。COBOL 登録集のなかの記述項は, 環境部, データ部もしくは手続き部, またはそれらを組み合わせたものの原始プログラム原文を含む。

(1) **水準1の特性** 登録集の水準1 (library level 1) は, 登録集から原始プログラムへ原文を複写する機能を有する。語単位の置き換えは, できない。

(2) **水準2の特性** 登録集の水準2 (library level 2) は, 登録集から原始プログラムの原文を複写する機能を有する。原文が複写される際, 語単位の置

き換えができる。

4. その他

JIS COBOL 原案は、工業技術院日本工業標準調査会の専門委員会で審議され、広く一般および識者からの意見をも参考にし、1972年春ごろに JIS COBOL として制定される予定である。

JIS COBOL 原案についての意見または質問は、上記の委員会または本学会まで文書で寄せていただいた

い。原案作成過程についての資料は、学会に残部がある模様である。なお本稿の執筆は、幹事伊藤靖彦（日電）が行なった。

参考文献

情報処理学会：電子計算機の国際標準化，1971。

（昭和46年4月16日受付）

ニ ュ ー ス

CDC 6600 センター

世界最大級のコンピュータ CDC 6600 システム（記憶容量 131K 語，1 語 60 ビット構成）による計算センターが稼働をはじめた。

これは、伊藤忠電子計算サービス（株）が東京・日本橋本町 2-6-4 大阪合同ビル 1 階に開放したもので、同社はさる 41 年 2 月には、大型コンピュータ CDC 3600 システム（記憶容量 98K 語，1 語 48 ビット構成）による計算センターを開設しており、その情報処理能力はわが国最大級の規模となった。

CDC 6600 システムは、1 台のセントラル・プロセッサと、10 台のペリフェラル・プロセッサの合計 11 台の独立したコンピュータがひとつのシステムを構成し、超高速の多重レベルのデータ処理を行なうことができる。また、オペレーティング・システム（OS）は SCOPE と呼ばれ、システム全体の効率のよいオペレーションを行なう。

ソフトウェアについては、科学技術計算、オペレーションズ・リサーチ（OR）、各種の経営計算、事務データ処理などの各分野にわたる豊富なアプリケーションが整備されているほか伊藤忠電子計算サービス（株）が従来独自に開発したプログラムの多数が CDC 6600 用に変換されている。

CDC 6600 センターの利用については、独特の利用規程がつくられ、たとえば、利用料金には“優先度制”を採用し、ユーザーの緊急度に応じた料金を適用することになっている。

また、同社は CDC 6600 センターの利用にあたってのプログラム講習会を随時開催しているほか、「CDC 6600 プログラム一覧表」なども刊行し、ユーザーの便宜をはかっている。利用についての問い合わせ先は、

東京都中央区日本橋本町 2-6-4 大阪合同ビル
電話 (663) 4211 (代) カストマ・サービス部

「プログラムの基礎理論」シンポジウム

3 月 22 日～24 日の 3 日間、京都大学数理解析研究所に 63 名が参集して、表記の研究集会が行なわれた。今回はプログラムの理論 (Mathematical Theory of Computation) の最近の発展の中でも最も重要な地位をしめる 2 つと思われる R. W. Floyd のプログラムの「正当性」の概念に基づく諸結果および D. Scott のプログラムの意味論と証明論での束論的考察の紹介（講演 2 および 3）と、4 件の新しい研究結果の発表とが行なわれ、活発な討論が行なわれた。

ソフトウェアの専門家、数学基礎論の研究者、大学院学生のように若い人達もそれぞれの立場からききわめて活発に討論に加わり、この分野の今後の発展を想像させられた。とくに「正当性」の証明については、それがプログラミングでのこれまでの唯一つの実際的手段であった「デバッキング」を補う可能性をもっているために議論が集中した。

講演題目は次のとおりである（敬称略）。なお、講演内容は近く数理解析研究所講究録として刊行されることになっている。

- 1) 謝 章文：メタフローを持つプログラム形について
- 2) 寛 捷彦：Tree-Sort アルゴリズムの正当性の証明について
- 3) 小野寛晴：Mathematical Theory of Computation における D. Scott の方法
- 4) 金山 裕：束準同形としてのプログラムと方程式解の一貫性
- 5) 五十嵐滋：ALGOL 型プログラム言語における意味論と証明論
- 6) 鈴木則久：プログラムの正当性を機械的に証明するための一方法

* * *