

情報処理基礎講座

COBOL応用編

監修 工学博士 松尾三郎

電子開発学園

謝 辞

本テキストは、データ組織言語協議会（CODASYL）のCOBOL委員会とヨーロッパ計算機製造者協会（ECMA）との努力によって開発されたCOBOLに基づき、その要請にしたがってつぎの謝辞を掲げます。

「いかなる組織体も自由に教育用説明書その他の目的で、COBOL文法説明書とその仕様の一部または全部を複製し、あるいは報告書中の着想を利用してかまいません。ただし、その文書の序文の一部にこの部分を掲載しなければなりません。書評などで短い文章を引用するときは、出典についての謝辞にCOBOLの名称を挙げれば全文を掲げる必要はありません。

COBOLは、産業界の言語であって、いかなる会社、会社団体、組織または組織団体の所有物でもありません。

COBOL言語作成の寄与者またはCOBOL委員会は、プログラミング組織と言語の正確さ、機能についていかなる保証をもしません。また、これに関する事項について、いかなる責任をも負いません。

COBOL保全の手続きは確立しています。変更を提案する手続きについては、データ組織言語会議（CODASYL）の理事会に照会して下さい。

ここで利用した以下の資料の著書および著作権者は、COBOL仕様書にその一部または全部を利用することを認可しています。このことは、プログラミング説明書や類似の出版物にCOBOL仕様書を複製、利用する場合にも適用されます。

FLOW-MATIC（スペリランド社の商標）、

Programming for Univac (R) I and II, Data Automation systems,

スペリランド社, 1958年, 1959年版権；

IBM Commercial Translator,

図書番号 F 28-8013, IBM社, 1959年版権；

FACT,

DSI 27A5260-2760, ミネアポリス・ハニウエル社, 1960年版権；」

本テキストの作成者は、CODASYL COBOL委員会各寄与者の努力と成果を高く評価し、ここに感謝の意を表します。

はじめに

●本書の特徴

昨今のコンピュータの普及には目を見張るものがあります。装置も次第に小型化し、いわゆるオフィスオートメーション（OA）化が全国的にかなり浸透してきました。これらに伴い、従来コンピュータとは縁のなかった一般の人々でも、コンピュータを扱わなければならない状況になってきています。また、コンピュータの新機種が次々と登場する一方、事務処理用の共通言語である「JIS COBOL」にも改正が重ねられてきました。

このような情勢をふまえ、本書は電子開発学園の長い歴史と豊富な教育経験を基に、COBOLテキストの応用編として、COBOLの基礎機能を学習した方を対象に編集したものです。

「COBOL応用機能編」初版を発行し、以来いろいろな情報を取り入れながら版を重ねてきましたが、その間の情勢の急激な変化に伴い、内容的に少々不備な点が目立ってまいりました。そこで、このたび現在の情勢を念頭におき、それらを十分に反映させたテキストとして、内容を全面的に見直し、新版として発行する運びとなりました。

本書は改版前と同様のねらいをもって編集し、COBOLの応用機能をわかりやすく解説してあります。具体的には次のような特徴を持っています。

- (1) 文法の説明はとかく堅苦しくなりがちなため、説明のつど使用例を挙げ、理解しやすいよう配慮しました。また、章の最後には練習問題を用意し、理解を確たるものにするよう配慮しました。
- (2) COBOLを学習するに当たっては、コンピュータを実際に使いながら少しずつ身につけていくのが効果的です。いちどにあれこれと見ってしまうと知識の混乱を招くばかりか、理解があやふやなものになってしまいます。

このため、「COBOL基礎編」では、COBOL文法の基礎を段階的に学習できるようにしました。

これをうけて、本書では、拡張機能を機能ごとにまとめ、「COBOL基礎編」を学習済みの方なら、どの章からでも学習できるように配慮しました。

●本書を使用するに当たって

本書は、基本的には、「JIS X3002(1992) 電子計算機プログラム言語COBOL」に準拠しているため、個々の計算機によるCOBOL文法の相違点については触れていません。ただし、本文中のプログラム例の装置名は、MS-DOSに準拠した形にしています。

また、本書は、COBOL文法の拡張機能を解説したものですが、あまり使用されないと思われる機能については省略しました。

省略した機能は下記のとおりです。

・見出し部

AUTHOR段落,
INSTALLATION段落,
DATE-WRITTEN段落,
DATE-COMPILED段落,
SECURITY段落,

・環境部

構成節のSOURCE-COMPUTER段落のDEBUGGING MODE指定,
構成節のOBJECT-COMPUTER段落のMEMORY SIZE指定とSEQUENCE指定と
SEGMENT-LIMIT指定,
構成節のSPECIAL-NAMES段落,
入出力節のI-O-CONTROL段落,
ファイル管理記述項のOPTIONAL指定とRESERVE指定とPADDING CHARACTER
指定とRECORD DELIMITER指定,

・データ部

連絡節,
ファイル記述項のVALUE OF指定とCODE-SET指定,
整列併合用ファイル記述項のRECORD CONTAINS句とDATA RECORD句,
データ記述項のSIGN指定とSYNC指定とJUSTIFIED指定とBLANK ZERO指定,

・手続き部

区分番号,

ACCEPT文のFROM指定,

ALTER文,

CANCEL文,

CLOSE文のREEL指定, UNIT指定, WITH NO REWIND指定, WITH LOCK指定,

DISABLE文,

DISPLAY文のUPON指定とNO ADVANCING指定,

ENABLE文,

ENTER文,

EXIT文,

PURGE文,

RECEIVE文,

REPLACE文,

SEND文,

SET文の呼び名と条件名,

SORT文のWITH DUPLICATES指定とCOLLATING SEQUENCE指定,

STOP文の定数,

USE文のGLOBAL指定

目 次

謝 辞

はじめに

第1章 表操作	1
1.1 概要	1
1.2 逐次表引き	1
1.2.1 データ部の記述	1
(1) OCCURS句	1
1.2.2 手続き部の記述	2
(1) SET文	2
(2) SEARCH文	5
1.2.3 例題	9
1.3 二分表引き	12
1.3.1 手続き部の記述	12
(1) SEARCH文	12
1.3.2 例題	14
1.4 練習問題	16
第2章 整列併合機能	27
2.1 概要	27
2.2 整列併合機能に関する記述項	28
2.2.1 SELECT句	28
2.2.2 ファイル節	28
(1) 概要	28
(2) 整列併合用ファイル記述項	29
2.3 整列機能	30
2.3.1 SORT文	30
2.3.2 RELEASE文	34
2.3.3 RETURN文	34
2.3.4 プログラム例	36
2.4 例題	38
2.5 練習問題	40
2.6 併合機能	44
2.6.1 MERGE文	44

2.6.2	プログラム例	47
第3章	プログラム間連絡機能	49
3.1	概要	49
3.2	プログラム間連絡	50
3.2.1	制御の移行	50
3.2.2	パラメタの受渡し	51
3.2.3	呼ぶプログラム	52
(1)	CALL文	52
3.2.4	呼ばれるプログラム	54
(1)	連絡節	54
(2)	手続き部の見出し	55
(3)	EXIT PROGRAM文	55
3.3	原始プログラムの入れ子	56
3.3.1	うちプログラム	56
3.3.2	うちプログラムの分類	57
3.3.3	見出し部の記述	57
(1)	プログラム名段落	57
3.4	共通データや共通ファイルの参照	58
3.4.1	GLOBAL句	58
3.4.2	EXTERNAL句	58
3.5	例題	60
3.6	練習問題	72
第4章	索引ファイル	75
4.1	概要	75
4.2	呼出し法	76
4.3	環境部の記述	77
4.3.1	SELECT句	77
4.4	手続き部の記述	80
4.4.1	OPEN文	80
4.4.2	CLOSE文	82
4.4.3	READ文	82
4.4.4	WRITE文	87
4.4.5	REWRITE文	91

4.4.6	START文	95
4.4.7	DELETE文	98
4.4.8	呼出し法とOPEN文のモードによる入出力の方法	102
	(1) 順呼出し法	102
	(2) 乱呼出し法	102
	(3) 動的呼出し法の順呼出し	102
	(4) 動的呼出し法の乱呼出し	103
4.4.9	USE文	104
4.4.10	入出力状態	106
	(1) 概要	106
	(2) 入出力状態の値	106
4.5	例題	109
4.6	練習問題	115
第5章	相対ファイル	123
5.1	概要	123
5.2	呼出し法	124
5.3	環境部の記述	124
	5.3.1 SELECT句	124
5.4	手続き部の記述	126
	5.4.1 OPEN文	126
	5.4.2 CLOSE文	128
	5.4.3 READ文	128
	5.4.4 WRITE文	133
	5.4.5 REWRITE文	137
	5.4.6 START文	140
	5.4.7 DELETE文	143
	5.4.8 呼出し法とOPEN文のモードによる入出力の方法	147
	(1) 順呼出し法	147
	(2) 乱呼出し法	147
	(3) 動的呼出し法の順呼出し	147
	(4) 動的呼出し法の乱呼出し	148
	5.4.9 USE文	149
	5.4.10 入出力状態	151
	(1) 概要	151

(2) 入出力状態の値	151
5.5 例題	154
5.6 練習問題	162
第6章 報告書作成機能	171
6.1 概要	171
6.2 報告書の構造	171
6.3 データ部の記述	173
6.3.1 ファイル記述項	173
6.3.2 報告書節	174
(1) 概要	174
(2) 報告書記述項	174
(3) 報告集団記述項	177
6.3.3 特殊レジスタ	183
(1) PAGE-COUNTER	183
(2) LINE-COUNTER	183
6.4 手続き部の記述	184
6.4.1 INITIATE文	184
6.4.2 GENERATE文	184
6.4.3 TERMINATE文	186
6.5 例題	188
6.6 練習問題	208
第7章 原始文操作機能	215
7.1 概要	215
7.2 COPY文	215
7.3 例題	217
付 録	221

第1章 表操作

1.1 概要

表引きについては基礎機能編で学習済みであるが、ここでは表引きを自動的に行うための文とその表の定義法について学習する。

1.2 逐次表引き

表の各要素を順番に探索することを、逐次表引きという。この方法は、表の要素数が少ない場合や表の探索項目が不規則な並びをしている場合に有効な方法である。

1.2.1 データ部の記述

(1) OCCURS句

OCCURS（反復）句は、同じデータ構造が繰り返される時、その記述項をいちいち指定する手間を省くためのものであった。（基礎機能編で学習済み）

ここでは、表の要素を参照または更新するときの指標を定義するINDEXED BY指定や表の並び方を記述するKEY IS指定について学習する。

●一般形式

OCCURS 整数 2 TIMES

{ ASCENDING } KEY IS {データ名2} ...
{ DESCENDING }

[INDEXED BY {指標名1} ...

規 則

- ① 整数 2 は、同じ形式のデータ項目の反復回数を表す。
- ② KEY IS指定でASCENDINGを書いたとき、その表はデータ名2の値の昇順に並んでいなければならない。
- ③ KEY IS指定でDESCENDINGを書いたとき、その表はデータ名2の値の降順に並んでいなければならない。
- ④ INDEXED BY指定を書くと、この記述項およびこの記述項に従属する項目を指標付けによって参照することができる。指標の定義は完全にハードウェアに依存しているので、それを記述する独立した記述項はない。

<例>

8	12
01	T-TBL.
	02 T-MEISAI OCCURS 5 ASCENDING T-NO INDEXED IDX.
	03 T-NO PIC X(2).
	03 T-NAME PIC X(10).

1.2.2 手続き部の記述

(1) SET文

表引きの添字としてデータ項目を使用する場合には、その初期値設定や値の変更には転記文や算術文を利用した。しかし、指標名による添字付けの場合は、SET（設定）文を利用する。

SET（設定）文は、表要素の指標を設定して、表要素の参照点を定める。

●一般形式

<p>書き方1</p> $\text{SET } \left\{ \begin{array}{l} \text{指標名1} \\ \text{一意名1} \end{array} \right\} \cdots \text{TO } \left\{ \begin{array}{l} \text{指標名2} \\ \text{一意名2} \\ \text{整数1} \end{array} \right\}$ <p>書き方2</p> $\text{SET } \{ \text{指標名3} \} \cdots \left\{ \begin{array}{l} \text{UP BY} \\ \text{DOWN BY} \end{array} \right\} \left\{ \begin{array}{l} \text{一意名3} \\ \text{整数2} \end{array} \right\}$ <p>書き方3</p> $\text{SET } \{ \text{条件名1} \} \cdots \text{TO TRUE}$

規 則

- ① 一意名1および一意名2は、指標データ項目または整数項目でなければならない。
- ② 一意名3は、整数項目でなければならない。
- ③ 整数1および整数2は符号を付けてもよいが、整数1は正の整数でなければならない。
- ④ 指標名は、OCCURS句のINDEXED BY指定に書くことによって、その表と関連付けられる。
- ⑤ 書き方1の動作は、次のいずれかになる。
 - ④ 指標名1または一意名1に、指標名2、一意名2または整数1の値を設定する。
 - ⑤ 一意名1が整数項目のときは、TOの後には指標名2だけを書くことができる。
 - ⑥ 指標名1、一意名1の繰返しが書いてあれば、処理を繰り返す。
- ⑥ 書き方2では、指標名3の内容が、整数2または一意名3のデータ項目の値だけ増やされる（UP BY）か減らされる（DOWN BY）。さらに、指標名3の繰返しに対して処理を繰り返す。
- ⑦ 書き方1のSET文に対する各作用対象の組合せの可否を表1.1に示す。
- ⑧ 書き方3では、条件名1に関連するVALUE句の定数を、VALUE句の規則に従って、条件変数に設定する。VALUE句に2つ以上の定数を書いたときは、そのVALUE句に書いた最初の定数の値が条件変数に設定される。

表 1.1 SET文における各作用対象の組合せの可否

送出し側	受取り側		
	整数データ項目	指 標 名	指標データ項目
整数	—	○	—
整数データ項目	—	○	—
指標名	○	○	○
指標データ項目	—	○	○

(注) ○は、その組合せが可能なことを、—は不可能なことを示している。

<例1> 指標名IDX 1 に一意名 I の値を設定する。

8	12	
		SET IDX1 TO I.

<例2> 指標名IDX 2 に整数の値10を設定する。

8	12	
		SET IDX2 TO 10.

<例3> 指標名IDX 3 に指標名IDX 4 の値を設定する。

8	12	
		SET IDX3 TO IDX4.

<例4> 一意名 J に指標名IDX 5 の値を設定する。

8	12	
		SET J TO IDX5.

<例5> 指標名IDX 6 の値を2だけ増やす。

8	12	
		SET IDX6 UP BY 2.

<例6> 指標名IDX 7 の値を一意名Kの値だけ減らす。

8	12	
		SET IDX7 DOWN BY K.

<例7> 条件変数CDN 1 にVALUE句で書いた値（"ON"）を設定する。

8	12	
01		EOF-SW PIC X(3) VALUE "OFF".
88		CDN1 VALUE "ON".
		⋮
		SET CDN1 TO TRUE.

(2) SEARCH文

SEARCH（表引き）文は、指定した条件を満足する表要素を探し、対応する指標の値がその表要素を指すようにする。

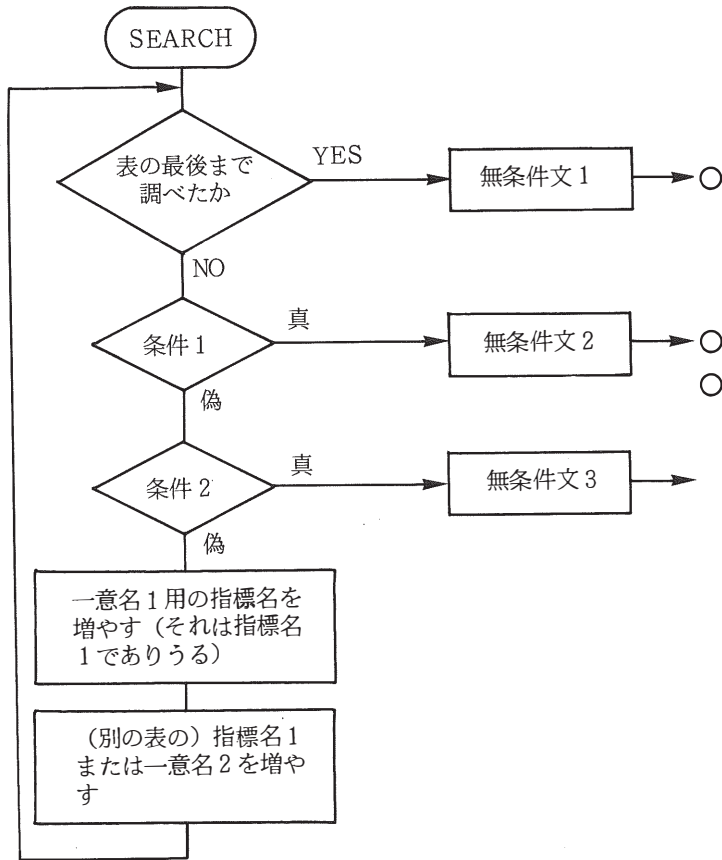
●一般形式

$\text{SEARCH 一意名 1 } \left\{ \begin{array}{l} \text{VARYING } \left\{ \begin{array}{l} \text{一意名 2} \\ \text{指標名 1} \end{array} \right\} \\ \\ \text{[AT END 無条件文 1]} \\ \\ \left\{ \begin{array}{l} \text{WHEN 条件 1} \\ \text{NEXT SENTENCE} \end{array} \right\} \dots \\ \\ \text{[END-SEARCH]}$
--

規 則

- ① 一意名 1 は、添字や部分参照を付けてはならない。その一意名 1 の記述には、INDEXED BY指定をもつOCCURS句がなければならない。
- ② 一意名 2 は、指標データ項目または整数項目でなければならない。
- ③ 条件 1 は、どのような条件式であってもよい。
- ④ SEARCH文の範囲は、次のいずれかで終わる。
 - Ⓐ 入れ子の同じレベルにあるEND-SEARCH指定。
 - Ⓑ 分離符の終止符。
 - Ⓒ 直前のIF文に対応するELSE指定またはEND-IF指定。
- ⑤ SEARCH文は、そのときの指標の値から始める逐次表引き操作を行う。
 - Ⓐ SEARCH文の実行を始めたとき、一意名 1 用の指標名に一意名 1 の最高の出現番号より大きい値が入っていると、表引きは直ちに終わりとなる。AT END指定があればその無条件文 1 を実行し、なければ制御はSEARCH文の終わりに移る。
 - Ⓑ SEARCH文の実行を始めたとき、一意名 1 用の指標名に一意名 1 の最高の出現番号以下の値が入っていると、SEARCH文は、次の動作を行う。
 - ・SEARCH文は、書いてある順に条件を検査する。検査される項目の番号として、そのときの指標名の値を用いる。
 - ・どの条件も満足されないときは、一意名 1 用の指標名の値を増やして次の要素が参照できるようにする。
 - ・指標名の新しい値を使って、上の処理を繰り返す。
 - ・一意名 1 用の指標名の値が表の範囲外の出現番号になると、上のⒶで述べたような終わり方をする。

- ・条件を検査していて、条件の1つが満足されると、表引きはすぐ終わりとなり、その条件に伴う無条件文に制御を移す。NEXT SENTENCE指定が書かれていれば次の実行完結文に移る。指標名には、この条件を満足させたときの出現番号が残る。
- ⑥ VARYING指定を書かないと、一意名1のOCCURS句中のINDEXED BY指定に書いた最初の指標名だけが、表引き操作に用いられる。
- ⑦ VARYING指定を書き、その句の指標名1が、一意名1のOCCURS句中のINDEXED BY指定に書いてある場合、この表引きにその指標名を使う。そうでない場合またはVARYING指定に一意名2を書く場合、一意名1のOCCURS句中のINDEXED BY指定で与えた最初の指標名だけを表引きに使う。この場合、さらに次の操作も行う。
 - ⑧ VARYING指定で指標名1を書き、その指標名1が別の表のOCCURS句中のINDEXED BY指定に書いてある場合、指標名1の表す出現番号は、一意名1用の指標名の表す出現番号と同じだけ同時に増加する。
 - ⑨ VARYING指定で一意名2を書き、その一意名2が指標データ項目の場合、その指標データ項目は、一意名1用の指標名の表す出現番号と同じだけ同時に増加する。一意名2が指標データ項目でない場合、そのデータ項目の値は、一意名1用の指標が増加するときに、1だけ増加する。
- ⑧ WHEN指定が2つある場合の流れ図は次のとおりとする。
- ⑨ END-SEARCH指定を書いた場合は、NEXT SENTENCE指定を書いてはならない。



<例1> 一意名T-MEISAIの表を先頭から調べ、一意名T-NOの値が一意名Nの値より大きい最初の要素の一意名T-NAMEの値を一意名NCに転記する。見つからなければ、空白を転記する。

8	12
01	NC PIC X(10).
01	N PIC 9(2) VALUE 5.
01	T-TBL.
	02 T-MEISAI OCCURS 5 INDEXED IDX.
	03 T-NO PIC 9(2).
	03 T-NAME PIC X(10).
	⋮
	SET IDX TO 1.
	SEARCH T-MEISAI AT END MOVE SPACE TO NC
	WHEN T-NO (IDX) > N
	MOVE T-NAME (IDX) TO NC
	END-SEARCH.

<例2> 一意名T-MEISAIの表を3番目の表要素から調べ、一意名T-NOの値が一意名Nの値以上かまたは一意名Mの値以下の最初の要素の一意名T-NAMEの値を一意名NCに転記する。見つからなければ、指標名IDXに9を設定する。

8	12
01	NC PIC X(10).
01	N PIC 9(2).
01	M PIC 9(2).
01	T-TBL.
02	T-MEISAI OCCURS 5 INDEXED IDX.
03	T-NO PIC 9(2).
03	T-NAME PIC X(10).
⋮	⋮
MOVE 5	TO N.
MOVE 2	TO M.
SET	IDX TO 3.
SEARCH	T-MEISAI AT END SET IDX TO 9
WHEN	T-NO (IDX) >= N
MOVE	T-NAME (IDX) TO NC
WHEN	T-NO (IDX) <= M
MOVE	T-NAME (IDX) TO NC
END-SEARCH.	

<例3> 複合条件を使用して<例2>と同じ処理を記述した例である。(手続き部のみ)

8	12
MOVE 5	TO N.
MOVE 2	TO M.
SET	IDX TO 3.
SEARCH	T-MEISAI AT END SET IDX TO 9
WHEN	T-NO (IDX) >= N
OR	T-NO (IDX) <= M
MOVE	T-NAME (IDX) TO NC
END-SEARCH.	

1.2.3 例題

<プロセス図>

```

    graph TD
      A[学生マスタ] --> B[探索]
      B --> C[探索表]
  
```

<処理> 学生マスタファイルを読み込み、クラス番号からクラス名を調べる。該当するクラス名があればその内容を印字する。該当するクラス番号がない場合にはエラーメッセージを表示する。

<ファイル構成>

ファイル	ファイル名	ファイル編成	ブロック化係数
学生マスタ	DSK-F	順	10
探索表	LIST-F	順	-

<入力形式> 学生マスタ

クラス番号	一連番号	データ部
X(3)	X(2)	X(75)

<出力形式> 探索表

```

    *** タンサウ ヒョウ ***
    クラス   バンゴウ       データ
    XXXX   XX   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....XXXXXXXXXXXXXXXXX
    :      :      :
    :      :      :
  
```

<対応表>

クラス番号	101	102	103	201	202	203	204	205	206
クラス名	S1A1	S1A2	S1A3	J1A1	J1A2	J1A3	J1A4	J1A5	J1A6

<コーディング>

```

1 IDENTIFICATION      DIVISION.
2 PROGRAM-ID.         EX1-2.
3*
4 ENVIRONMENT         DIVISION.
5 INPUT-OUTPUT       SECTION.
6 FILE-CONTROL.
7     SELECT DSK-F    ASSIGN "B:GAKUSEI.DAT".
8     SELECT LIST-F  ASSIGN "LST:".
9*
10 DATA              DIVISION.
11 FILE               SECTION.
12 FD DSK-F
13     BLOCK 10 RECORDS.
14 01 DSK-R.
15     02 D-CLASS    PIC X(3).
16     02 D-BANGO   PIC X(2).
17     02 D-DATA    PIC X(75).
18 FD LIST-F.
19 01 LIST-R        PIC X(132).
20 WORKING-STORAGE   SECTION.
21 01 LCNT          PIC 9(2) VALUE 30.
22 01 MIDASHI-1.
23     02           PIC X(33) VALUE SPACE.
24     02           PIC X(19) VALUE "*** タンサカ ヒヨウ ***".
25 01 MIDASHI-2.
26     02           PIC X(10) VALUE SPACE.
27     02           PIC X(12) VALUE "クラス ハンゴウ".
28     02           PIC X(30) VALUE SPACE.
29     02           PIC X(4)  VALUE "デ-タ".
30 01 MEISAI-R.
31     02           PIC X(9)  VALUE SPACE.
32     02 M-CLASS   PIC X(4).
33     02           PIC X(5)  VALUE SPACE.
34     02 M-BANGO  PIC X(2).
35     02           PIC X(10) VALUE SPACE.
36     02 M-DATA   PIC X(75).
37 01 TAIYOU-HYO.
38     02           PIC X(7)  VALUE "101S1A1".
39     02           PIC X(7)  VALUE "102S1A2".
40     02           PIC X(7)  VALUE "103S1A3".
41     02           PIC X(7)  VALUE "201J1A1".
42     02           PIC X(7)  VALUE "202J1A2".
43     02           PIC X(7)  VALUE "203J1A3".
44     02           PIC X(7)  VALUE "204J1A4".
45     02           PIC X(7)  VALUE "205J1A5".
46     02           PIC X(7)  VALUE "206J1A6".
47 01 T-HYO REDEFINES TAIYOU-HYO.
48     02 CLASS-HYO OCCURS 9 INDEXED IDX.
49         03 C-NO   PIC X(3).
50         03 C-NAME PIC X(4).

```

```

51 01 ERROR-MESSAGE.
52 02 ERR-CLASS PIC X(7).
53 02 PIC X(22) VALUE "*** ガ`イウ クラス ナシ ***".
54*
55 PROCEDURE DIVISION.
56 HAJIME.
57 OPEN INPUT DSK-F OUTPUT LIST-F.
58 READ DSK-F AT END MOVE HIGH-VALUE TO D-CLASS.
59 PERFORM UNTIL D-CLASS = HIGH-VALUE
60     SET IDX TO 1
61     SEARCH CLASS-HYO AT END SET IDX TO 99
62     WHEN C-NO (IDX) = D-CLASS
63         MOVE C-NAME (IDX) TO M-CLASS
64     END-SEARCH
65     IF IDX = 99
66     THEN
67         MOVE D-CLASS TO ERR-CLASS
68         DISPLAY ERROR-MESSAGE
69     ELSE
70         MOVE D-BANGO TO M-BANGO
71         MOVE D-DATA TO M-DATA
72         PERFORM PRINT-RTN
73     END-IF
74     READ DSK-F AT END MOVE HIGH-VALUE TO D-CLASS
75     END-READ
76 END-PERFORM.
77 CLOSE DSK-F LIST-F.
78 STOP RUN.
79*
80 PRINT-RTN.
81     IF LCNT >= 30
82     THEN
83         MOVE SPACE TO LIST-R
84         WRITE LIST-R AFTER PAGE
85         WRITE LIST-R FROM MIDASHI-1 AFTER 2
86         WRITE LIST-R FROM MIDASHI-2 AFTER 2
87         MOVE 0 TO LCNT
88     END-IF.
89     WRITE LIST-R FROM MEISAI-R AFTER 2
90     ADD 1 TO LCNT

```

1.3 二分表引き

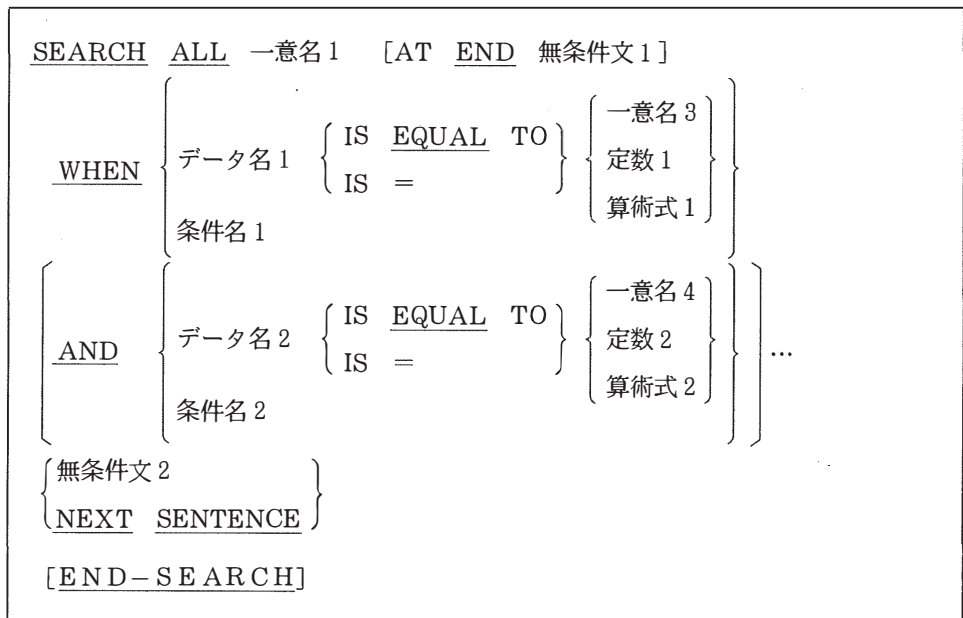
表の各要素がある項目の値の昇順または降順に並んでいる場合に、探索範囲を半分、半分…とせばめながら探索することを、二分（非逐次）表引きという。この方法は、表の要素数が多いときでも、探索項目が規則的な並びをしているならば効率的に探索できる方法である。

1.3.1 手続き部の記述

(1) SEARCH文

SEARCH（表引き）文は、指定した条件を満足する表要素を探し、対応する指標の値がその表要素を指すようにする。

●一般形式（書き方2）



規 則

- ① 一意名1は、添字や部分参照を付けてはならない。その一意名の記述には、INDEXED BY指定とKEY IS指定をもつOCCURS句がなければならない。
- ② データ名1、データ名2は、一意名1のOCCURS句のKEY IS指定の中に書かれていなければならない。
- ③ 一意名1用の指標名の当初の値は無視される。その値は、作成者の定めた方法によって変化する。その値は、表の最後の要素の番号の値より大きくなることもないし、表の最初の要素の番号の値より小さくなることもない。
- ④ 上記の範囲内のどの指標の値をとっても、WHEN指定のいずれの条件も満足しないとき、制御はAT END指定があれば無条件文1に移り、なければSEARCH文の

終わりに移る。この場合には、指標の最後の値は規定されていない。

- ⑤ すべての条件が満足されたときは、指標に条件を満足させた要素の番号の値が入って制御は無条件文2に移る。NEXT SENTENCE指定が書かれていれば、制御は次の実行完結文に移る。
- ⑥ 無条件文1または無条件文2がGO TO文で終わっていないとき、その無条件文を実行した後、制御はSEARCH文の終わりに移る。
- ⑦ END-SEARCH指定を書いた場合は、NEXT SENTENCE指定を書いてはならない。

<例1> 一意名T-MEISAIの表を非逐次に調べ、一意名W-NOの値と一致するデータ名T-NOの要素を探し、データ名T-NAMEの要素のうちその出現番号と同じ出現番号をもつ要素の値を一意名W-NAMEに転記する。見つからなければ、空白を転記する。

8	12
01	W-NO PIC X(2).
01	W-NAME PIC X(10).
01	T-TBL.
	02 T-MEISAI OCCURS 5
	ASCENDING T-NO
	INDEXED IDX.
	03 T-NO PIC X(2).
	03 T-NAME PIC X(10).
	⋮
	ACCEP T W-NO.
	SEARCH ALL T-MEISAI AT END MOVE SPACE TO W-NAME
	WHEN T-NO (IDX) = W-NO
	MOVE T-NAME (IDX) TO W-NAME
	END-SEARCH.

1.3.2 例題

1.2.3の例題では、逐次表引きを使用した。ここでは、同じ例題を二分表引きを使用してコーディングしてみよう。

<コーディング>

```

1 IDENTIFICATION      DIVISION.
2 PROGRAM-ID.         EX1-3.
3*
4 ENVIRONMENT         DIVISION.
5 INPUT-OUTPUT       SECTION.
6 FILE-CONTROL.
7   SELECT DSK-F     ASSIGN "B:GAKUSEI.DAT".
8   SELECT LIST-F   ASSIGN "LST:".
9*
10 DATA              DIVISION.
11 FILE              SECTION.
12 FD DSK-F
13   BLOCK 10 RECORDS.
14 01 DSK-R.
15   02 D-CLASS     PIC X(3).
16   02 D-BANGO    PIC X(2).
17   02 D-DATA     PIC X(75).
18 FD LIST-F.
19 01 LIST-R       PIC X(132).
20 WORKING-STORAGE   SECTION.
21 01 LCNT        PIC 9(2) VALUE 30.
22 01 MIDASHI-1.
23   02           PIC X(33) VALUE SPACE.
24   02           PIC X(19) VALUE "*** タシヤク ヒヨウ ***".
25 01 MIDASHI-2.
26   02           PIC X(10) VALUE SPACE.
27   02           PIC X(12) VALUE "クラス バンゴウ".
28   02           PIC X(30) VALUE SPACE.
29   02           PIC X(4) VALUE "テ-タ".
30 01 MEISAI-R.
31   02           PIC X(9) VALUE SPACE.
32   02 M-CLASS    PIC X(4).
33   02           PIC X(5) VALUE SPACE.
34   02 M-BANGO   PIC X(2).
35   02           PIC X(10) VALUE SPACE.
36   02 M-DATA    PIC X(75).
37 01 TAIYOU-HYO.
38   02           PIC X(7) VALUE "101S1A1".
39   02           PIC X(7) VALUE "102S1A2".
40   02           PIC X(7) VALUE "103S1A3".
41   02           PIC X(7) VALUE "201J1A1".
42   02           PIC X(7) VALUE "202J1A2".
43   02           PIC X(7) VALUE "203J1A3".
44   02           PIC X(7) VALUE "204J1A4".
45   02           PIC X(7) VALUE "205J1A5".
46   02           PIC X(7) VALUE "206J1A6".

```

```

47 01 T-HYO REDEFINES TAIYOU-HYO.
48     02 CLASS-HYO OCCURS 9
49         ASCENDING C-NO
50         INDEXED IDX.
51         03 C-NO PIC X(3).
52         03 C-NAME PIC X(4).
53 01 ERROR-MESSAGE.
54     02 ERR-CLASS PIC X(7).
55     02 PIC X(22) VALUE "*** が 100 クラス 20 ***".
56*
57 PROCEDURE DIVISION.
58 HAJIME.
59 OPEN INPUT DSK-F OUTPUT LIST-F.
60 READ DSK-F AT END MOVE HIGH-VALUE TO D-CLASS.
61 PERFORM UNTIL D-CLASS = HIGH-VALUE
62     SEARCH ALL CLASS-HYO
63     AT END SET IDX TO 99
64     WHEN C-NO (IDX) = D-CLASS
65     MOVE C-NAME (IDX) TO M-CLASS
66 END-SEARCH
67 IF IDX = 99
68 THEN
69     MOVE D-CLASS TO ERR-CLASS
70     DISPLAY ERROR-MESSAGE
71 ELSE
72     MOVE D-BANGO TO M-BANGO
73     MOVE D-DATA TO M-DATA
74     PERFORM PRINT-RTN
75 END-IF
76 READ DSK-F AT END MOVE HIGH-VALUE TO D-CLASS
77 END-READ
78 END-PERFORM.
79 CLOSE DSK-F LIST-F.
80 STOP RUN.
81*
82 PRINT-RTN.
83 IF LCNT >= 30
84 THEN
85     MOVE SPACE TO LIST-R
86     WRITE LIST-R AFTER PAGE
87     WRITE LIST-R FROM MIDASHI-1 AFTER 2
88     WRITE LIST-R FROM MIDASHI-2 AFTER 2
89     MOVE 0 TO LCNT
90 END-IF.
91 WRITE LIST-R FROM MEISAI-R AFTER 2
92 ADD 1 TO LCNT

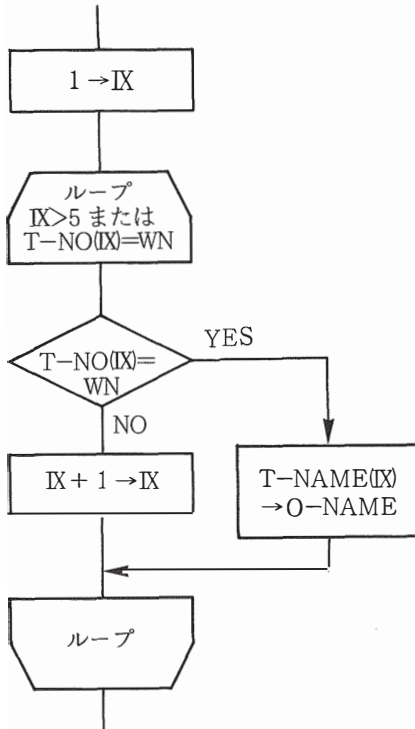
```

1.4 練習問題

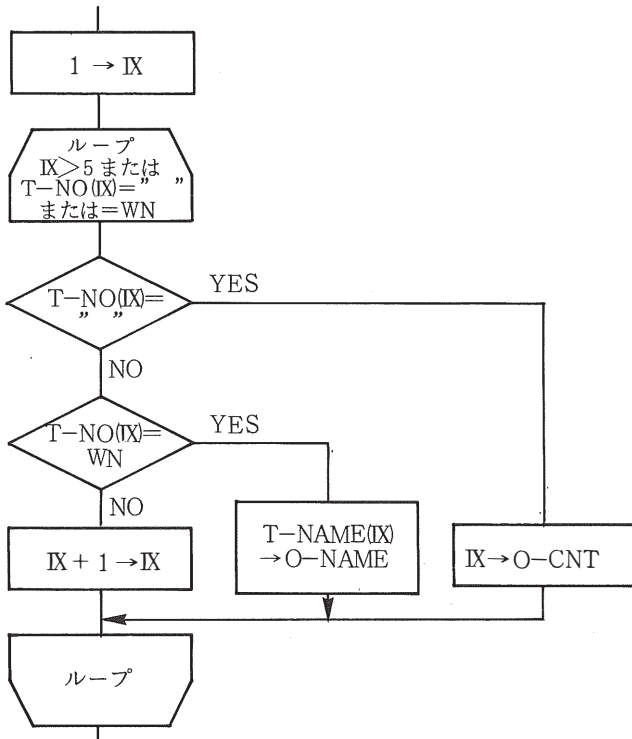
- (1) 次のように表および関連するデータ項目が定義されている。SEARCH文を用いて、(a)～(e)の流れ図をコーディングせよ。

8	12
01	WN PIC X(2).
01	O-NAME PIC X(20).
01	O-CNT PIC 9.
01	TBL-1.
	02 TBL-12 OCCURS 5 INDEXED IX.
	03 T-NO PIC X(2).
	03 T-NAME PIC X(20).

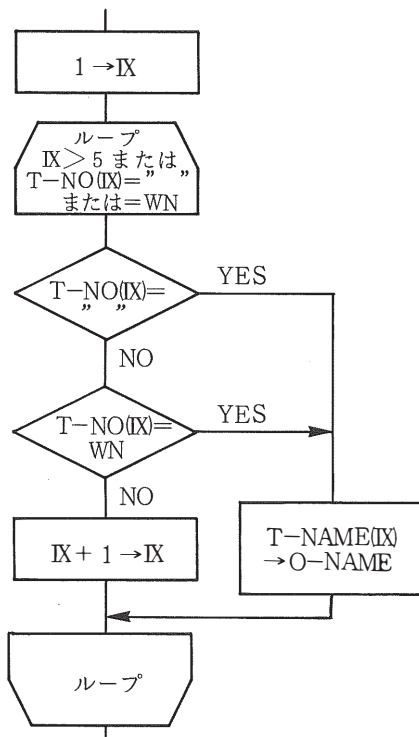
(a)



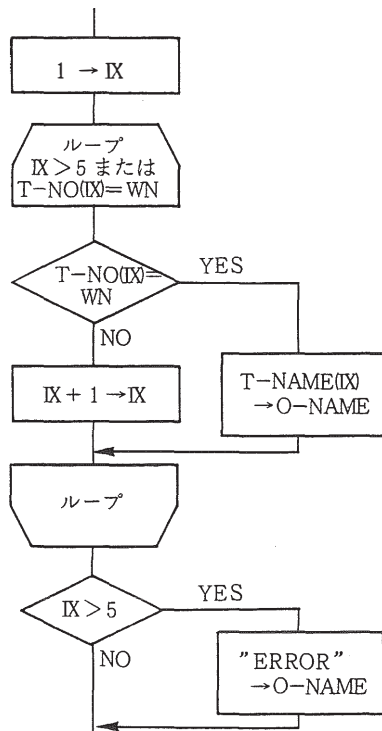
(b)



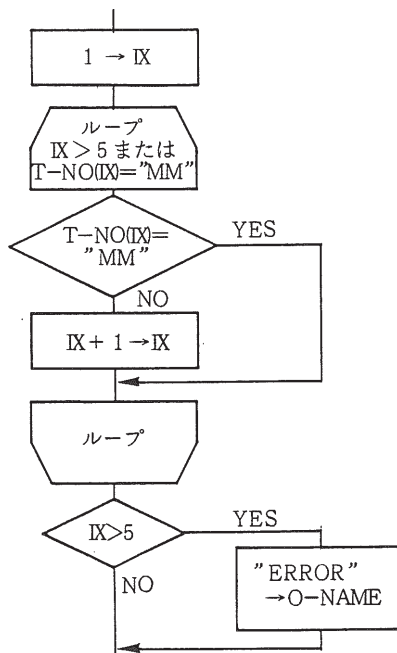
(c)



(d)



(e)



(f) (a)を二分探索で記述せよ。ただし、表はデータ名T-NOの要素の値の昇順に並べられているものとする。