

研究叢書 46

---

財務情報分析と新情報システム環境

民野 庄造 著

神戸大学

経済経営研究所

1995

研究叢書 46

---

財務情報分析と新情報システム環境

民野 庄造 著

神戸大学

経済経営研究所

1995

# 財務情報分析と新情報システム環境

民野庄造 著

神戸大学経済経営研究所

1995

# 財務情報分析と新情報システム環境

## 目 次

は し が き

謝 辞

### 第1章 プログラミング言語の構文記述とSECRETARY

— データ構造を中心に — .....	1
1.1 プログラミング言語の構文記法 .....	1
1.2 プログラミング言語とデータ構造 .....	4
1.2.1 プログラミング言語に見るデータ構造の沿革 .....	4
1.2.2 FORTRAN の構文について .....	7
1.2.3 C++言語の構文.....	12
1.2.3 a データ構造部分 .....	12
1.2.3 b 共通部分 .....	20
1.3 SECRETARYの構文記述 .....	24
1.4 プログラミング言語と情報分析システムの構文記述について .....	36
第2章 データ構造化の意義と新しい展開 .....	41
2.1 データ構造の意義と背景 .....	41
2.2 リスト構造について .....	43
2.3 リスト処理による財務諸表データの記述 .....	44
2.4 抽象データ型と財務諸表データ .....	51
2.4.1 抽象データ型について .....	52
2.4.2 財務諸表データのデータ構造 .....	55
2.4.3 新しいデータ型の課題 .....	60

2.5	新データ構造を組み入れた情報分析システム	61
第3章 財務データベース構築の基盤		
	— 収録データと管理・運用の基盤について —	65
3.1	データベースの発展と構築の要件	65
3.2	財務データベースの特色と収録データ	68
3.2.1	財務諸表データ	68
3.2.2	財務関連データ	72
3.2.3	財務データの特色	73
3.3	財務データベース利用マニュアル	74
3.4	財務分析の対象データと収録構造	79
第4章 新環境下の財務情報分析システム		
4.1	ネットワークと新しいマン/マシン・インターフェース	81
4.2	分散データベース	85
4.3	分散システムの構造	92
4.4	財務データベース分散システム	95
4.4.1	サーバ側ソフトウェア階層 (a-1 : DBa)	99
4.4.1 a	メイン・フレームDB管理 (s3)	99
	(1) 基幹データベースの役割と位置づけ	99
	(2) 分散システムと基幹データベース	102
	(3) 財務基幹データベース	111
4.4.1 b	財務情報分析とSECRETARY	114
4.4.1 c	OS (メイン・フレーム : s2)	117
4.4.2	サーバ側ソフトウェア階層 (a-2 : DB-b)	118
4.4.2 a	分散DBサーバ (s3 : DB-b)	119
	— 財務諸表作成支援, クライアント管理, 財務分析等 —	
4.4.2 b	分散システムのOS (s2,s1)	121

4.4.3 ユーザ側ソフトウェア階層 (b) .....	126
第5章 財務情報分析の計算・処理体系と情報分析システム .....	129
5.1 一般データ操作 (データ通覧・検索・編集加工・表出力等) .....	130
5.2 比率分析・百分比比率分析 .....	131
5.3 利子率関数に関連する分析 .....	134
5.4 損益分岐点分析とその同族 (資本回収点分析等) .....	141
5.5 財務分析における多変量解析 .....	146
5.6 財務分析・経営計画支援総合情報分析システム .....	159
5.7 財務諸表作成支援システム .....	162
5.8 その他の分析 (ポートフォリオ, CAPM等) .....	165
《付録》	
〈付録 1〉 経済経営情報分析システム (SECRETARY) の記述文法 .....	167
〈付録 2〉 財務分析問題 (第5章) 実行例 .....	245
〈付録 3〉 興銀財務データ項目 (DB名: 1部 KN1A, 2部 KN2A) .....	277
〈付録 4〉 COMPUS T A T財務データ項目 (DB名: CSSA) .....	281

## はしがき

現在のデジタル・コンピュータは、A.M.Turing (1912-1954) の論文「On Computable Numbers, with an Application to the Entscheidungsproblem」で計算の可能性が示され、同じく Turing によって考案された“チューリング・マシーン”と、John von Neumann (1903-1957) の発案とされる“プログラム内蔵方式 (Stored Program)”が基盤になっており、その型のコンピュータを“ノイマン型コンピュータ”と呼んでいる。その方式を取り入れて造られた最初の計算機は“EDSAC”で、ケンブリッジ大学の M.V.Wilkes が中心になり 1949 年 6月に完成し公開されたのに始まる。

以来約半世紀を経て発展してきた今日のコンピュータは、基本的にはノイマン型コンピュータの枠組みを超えるものではないがハードウェア、ソフトウェア、およびコンピュータを核とする情報システムの展開においてリニアな成長過程の中にも革新的出来事とその発展史に記録されている。

ハードウェアでは、素子の革新（高集積、高速スイッチング、パッケージング等）、記憶媒体（固体・磁気・光）の革新、生産技術の発展と、情報通信技術（ネットワーク）を挙げることが出来る。それらの発展により大容量記憶、高スループット、高コストパフォーマンスを持つ情報処理機械（資源）、および分散システム（クライアント／サーバ・システム）の開発基盤を情報システム開発者に提供することが出来た。

ソフトウェアでは、各種汎用プログラミング言語、データベース・システムの開発、人（利用者）の論理に近いオペレーティング・システム（OS : Macintosh, Windows, X Window 等）、国際規格・業界標準化の進展等共通の規範的なシステム開発環境が形成されることによってソフトウェアの命数の延長、流通を促進し、ソフトウェアの生産性向上、マルチベンダーによる情報システム構築等情報システム高度化への技術および制度的基盤が整備されてきた。

情報システムは、その時代々々のハードウェア／ソフトウェアの技術基盤を確認し開発・構築が進められる。その基盤の保証を得ない情報システムは、コンセプトのみ先行し実態のないものになることは歴史が証明している。

いま情報システムの世界で最も注目を集めているのはネットワークを基盤とする情報システムのシステムの展開である。光ファイバーによる高速データ伝送路、最近ではFDDI<sup>(1)</sup>、ATM<sup>(2)</sup>等による高速伝送路の結合・交換・制御等高度のネットワーク技術を取り入れた情報ネットワークシステムが研究用を中心に企業情報、商用サービスへと広がりを見せている。

この情報基盤の下では、各情報基地間、利用者のサイトと情報基地間のコストからみた情報伝達の経済距離は年々近くなり、また、リアルタイムの情報通信が可能になることにより、ネットワークを核とした新しい情報システムの設計を可能とする。

近来情報システムの集中化の限界から資源・処理機能の分散化を進めた分散処理システムが研究開発の大きなテーマになっており、実用化も進められているがそのバックボーンになっているのはネットワーク技術である。

さらに、高速データ伝送の進展と共に音声・画像等の各種メディアでの情報通信が可能になるに従って、情報システムのマネジメントから、あらゆる情報（分析）システム設計のコンセプトに至るまでネットワークを意識し、人と機械の自然なコミュニケーション、社会・生活に融合する情報システムの構築がシステム開発者に要求される。

本研究叢書は、これらの問題を念頭におき、データ処理の論理的な記述法に関する問題、ネットワーク環境下における情報システムの構築に関する問題、財務分析と情報分析システムの関連、等について考察する。

第1章は、人の扱う情報をコンピュータの論理に変換・蓄積・操作するための表現形式でデータ処理の永遠の研究テーマでもあるデータ形式とデータ構造について考える。具体的には、情報システムをコンピュータ上にシミュレート

させる「情報処理記述言語」についてデータ処理の骨格を規定する“データ構造記述機能”に視点をあて、汎用プログラミング言語を生み現在まで新時代の記述機能に対応してきた“FORTRAN”，抽象データ型の概念を汎用プログラミング言語に取り入れた“C++”，および我々が開発した経済経営情報分析システム“SECRETARY”に関して同一の表記法を用いて文法を記述しその構造と関連性を含め論理的な記述を試みる。

第2章は、データ構造の理論、開発されたデータ型（リスト、抽象データ型等）は適用分野（ここでは財務データ）に如何に実現されるかについて財務とデータ構造の両面から検討する。

第3章は、財務データベースについてシステムとして成立するための条件、基盤等に関する情報を提示する。

第4章は、近来急速に発展・展開されてきた情報ネットワークによって情報基盤が大きく変貌し、構築した情報システムの見直しをせまられている。

本章では、情報ネットワーク環境下における分散型財務情報分析システムの考え方、位置づけ、構造、新しいネットワーク環境に如何に組み入れられるか等について考究する。

第5章は、財務分析に関してその分析支援を行う情報分析システムに視点をあて、財務分析の主要な計算および処理体系のサーベイとそのアルゴリズムからの類型化を試み、それぞれの手法による分析例を示しながら財務情報分析の課題に接近する。

付録として、経済経営分野の情報分析に関して分析手続きの記述の自由度を上げ分析能率を上げる支援ツールとして我々が開発した“SECRETARY”の目的言語に関する記述文法を掲載する。

財務分析に限らず現代では、理論を実証・評価する問題、分析を必要とする問題は、実証（分析）ツール（コンピュータ、ネットワーク、ソフトウェア等）

の技術と強い相関で関連づけられている。ツールが時代を反映したものであれば実証研究はより進んだ理論を生みより速やかに課題に対しての解答を与えてくれるであろう。逆にツールが貧弱であることは実証研究に対して自己の責任を果たしていないことになる。コンピュータとネットワークが政治・社会・経済・産業・民生に浸透し急激に発展している現在、財務分析のシステム（ツール）といえども既存のシステムを見直し（アップデート）時代と共に歩むことは必要である。

### 《付記》

叢書を纏めるにあたって下記のことに関心した。

- (1) 論理・論述はヒューマンイズムに乖離していないか。
- (2) マクロを論じるときにミクロを、ミクロを論じるときにマクロを学習しているか。
- (3) 論述がハードウェア、またはソフトウェアに偏っていないか、体系的な取り組みと学際領域を見逃していないか。
- (4) 新しさをサーベイし、時代が求めているものを追求しているか。
- (5) 論述に関して主観と客観のバランスがとれ、史実・規範に対するアセスメントがされているか。

- 
- (1) FDDI (Fiber Digital Data Interface) : 光ファイバーをループ状に敷設しLANを結合するための標準プロトコル。
  - (2) ATM (Asynchronous Transfer Mode) : データ通信の非同期転送モードのことをいう。詳しくは、4.4.1 項の脚注を参照。

## 謝辞

本研究叢書の出版の機会を与えて頂いた経済経営研究所（以下同研究所）所長吉原英樹教授に感謝を申し上げます。

本書を纏めるにあたって全体の構成、記述の論点は何であるかについての有益な助言を頂いたこと、また筆者の専門外の財務に関する部分についてコメントを頂いた同研究所山地秀俊教授にお礼申し上げます。

私の専門領域である情報分析システムの研究・開発に際して有益なコメントを頂いた元同研究所片野彦二教授（現名古屋学院大学教授）、同じく下條哲司教授（現甲南大学教授）、同研究所阿部茂行教授、西島章次教授、井澤秀記助教授をはじめ同研究所の諸先生にお礼を申し上げたい。

伝統有る神戸大学金融研究会に門外の私を加えて頂いて以来現在まで教務に関しての励ましと機会を与えて頂いている元同研究所藤田正寛教授（現姫路獨協大学教授）に感謝を申し上げます。

元同研究所米花 稔教授（現福山大学教授）には私が同研究所に採用されて以来科研等のグループに入れて頂き退官されるまでの間ご教示を頂いた。

元同研究所定道 宏教授（現京都大学教授）には昭和46年頃より現在までの長い期間に渡り情報システム全般に関してのご指導を頂いた。

同研究所中野 勲教授には私の所属する機械計算室主任をされて以来現在に至る間いろいろな機会を与えて頂いたが応えることが出来なかった自分を恥お詫びしたい。

元同研究所藤希八郎先生（現大阪産業大学非常勤講師）には昭和36年より退官される昭和58年までの長期間同じ職場で勤務全般に渡りご指導を頂いた。

また、本業の機械計算室のサービス業務を持ちながら本叢書を纏めるためにその業務面の支援を頂いた元同研究所機械計算室の大村裕栄嬢、同研究所井口

美香助手にお礼を申し上げたい。

本書の出版に際しご支援を頂いた同研究所事務長海野興治氏，および編集でお世話になった同研究所研究助成室の戸村理恵助手をはじめみなさまにお礼を申し上げたい。

終わりに，情報システムに取り組むための基本的な指針を与えて頂き親切にご教示頂いた故小野二郎先生（元神戸大学教授）に拙書ではあるが本書を捧げ感謝の気持ちをあらわしたい。

# 第1章 プログラミング言語の構文記述 とSECRETARY

— データ構造を中心に —

## 1.1 プログラミング言語の構文記法

コンピュータの情報処理手続きを記述する汎用プログラミング言語，データベース・マネジメント・システム等の作成は，その言語の基準となる文法書を定めることから始められる。プログラミング等情報処理の記述は叙述方式をとり，基本的には文字・記号，またはその並びである文字・記号列の集合で個々の文が構成される。

プログラミング言語はこの文字列の集合にある規則を設け，データ処理に必要とされる諸機能の定義・意味づけを行い，それら全体を体系づけたものである。この文に関する各般の約束事を“構文規則”，あるいは“構文規約”と呼んでいる。

構文規則を作成（記述）し，そのドキュメントを客観的に表すには日本語による文を含め形式言語等何らかの言語を用いて説明しなければならない。

汎用プログラミング言語，DBMS（Data-Base Management System）のように汎用機能，複雑な文法書より成るシステムの構文を日本語のみで記述することは可能であるが，記述の曖昧さ，冗長性，規範的記述等の面で問題が残される。

プログラミング言語の構文記述に構文記法が取り入れられたのは“ALGOL 60”の文法書の記述に用いられた“BNF記法”<sup>(1.1)</sup>が最初である。

(1.1) BNF記法は，バックス（J.W. Backus）が提案し，ナウアー（P.Nauer）が“ALGOL 60”の文法書の記述のために編集した超言語。

この記法は、言語そのものを記述することから超言語（メタ言語：Meta Language）と呼ばれ、その言語によって記述・出力される言語を目的言語（Object-Language）と呼んでいる。

メタ言語を用いることにより膨大な文法規約を持つ形式言語の開発を容易にし、その構文全体を簡潔に説明・表記することが出来ると共に各システムの比較対照が容易になり、データ構造の発展を客観的に把握することが出来る。この構文記法の開発は、データ構造の理論、プログラミング言語の発展に大きく貢献した。

“BNF記法”は、“::=”，“〈”，“〉”，“|”等のメタ記号を使ってプログラミング言語の文法を帰納的に定義する。

たとえば、変数名は次のように表される。

〈変数名〉 ::= 〈英字〉 | 〈変数名〉 〈英数字〉

〈英数字〉 ::= 〈英字〉 | 〈数字〉

〈英字〉 ::= a | b | c | ..... | z

〈数字〉 ::= 0 | 1 | 2 | ..... | 9

- “::=”は左辺のメタ変数を右辺によって定義する記号。
- “|”は、右辺のメタ変数等を区切るメタ記号で、またはを意味する。
- “〈……〉”はメタ変数。
- A が空の場合は、次のように書かれる。

〈A〉 ::=

また、{ } 記号を用いその括弧内に書かれた形が0回以上繰り返されることを表すものとすれば、

〈変数名〉 ::= 〈英字〉 { 〈英数字〉 }

〈英数字〉 ::= 〈英字〉 | 〈数字〉

と簡単に表すことが出来る。

メタ言語による表記法は、BNF表記を基準にそれにいくつかの機能を補足・拡張した表記法も開発されている。“ALGOL N”の開発に用いられた“AN記法”はそのうちの1つである。

---

本叢書では、記述の便宜のため下記のメタ記号をBNF記法に追加して用いる。

《記》

- { } は、その括弧内の形が0回以上繰り返されることを表す。
  - (a) は、aの形であることを表す。結合の順序を表すために用いる。
  - [ ] は、その括弧内の形が空か、または括弧内の形かである。
  - ..... は、前に書かれている形が同様の規則で繰り返されることを表す。
  - 大文字（全角）の { , } , ( , ) , [ , ] , | はメタ記号として用いる。
  - 小文字（半角）の { , } , ( , ) , [ , ] , | はメタ定数で、プログラムの中ではそのままの形で書かれる。
- 

プログラミング言語の発展とともにそれに符合してプログラミング言語のデータ構造も変わりつつある。その動きの把握、サーベイの必要性も含めてポピュラーなプログラミング言語であるFORTRANの新版“FORTRAN 90”とオブジェクト指向プログラミングをリードしている“C++”を取り上げ新しいデータ構造の構文記述を中心に、“BNF記法”を用いて表記する。

記述の順序は、はじめに汎用プログラミング言語のデータ構造に関する新しい動きのサーベイとその構造についてのノート、続いて本経済経営研究所が開

- 
- (1.2) “汎用プログラミング言語”の“PL/I”, “C”に関してのBNF記法による記述は、本経済経営研究所刊、経済経営研究叢書・経営機械化シリーズ21, 「データ構造と財務諸表」で行った。

発した SECRETARY の言語構造の機能をメタ言語を用いて記述する。<sup>(1,2)</sup>

## 1.2 プログラミング言語とデータ構造

プログラミング言語の記述機能は二つの部分に大別して考えることが出来る。一つは問題を解くための計算・計算手続き等データ処理に関する記述を行う機能と、他方はデータ処理を効率的に行うためのデータ宣言・データ定義等データ記述機能である。説明の便宜上前者を“アルゴリズムの記述”，後者を“データ構造記述”と呼ぶことにする。ここではプログラミング言語のデータ構造記述に関して、以後のプログラミング言語に影響を与えたデータの概念・方式・データ型等を言語の発展史として捉えてみることにする。

### 1.2.1 プログラミング言語に見るデータ構造の沿革

プログラム言語の初期につくられた記号アドレス方式言語（アセンブリ言語）はそれまで使われてきた機械語プログラムの命令語・記憶域のアドレス・データを記号で指定できるようにしプログラムの生産性を上げプログラムの変更を容易にした画期的な記述言語であるがデータに関する扱いはハードウェアに依存したもので、そこでは抽象的な意味でのデータの型、データ定義という概念は取り入れられていない。

データ型、データ構造（配列）の概念が最初に取り入れられたのはコンパイラで、その最初は 1956年 IBMのバックスラによって開発された IBM-704 型コンピュータ用 FORTRAN であり汎用プログラミング言語として現在でも科学計算分野では最もよく使われている。

この原型の FORTRAN はデータ定義に関するステートメント“DIMENSION”，“EQUIVALENCE”を持ちデータ型の基本となった配列の考え方が

取り入れられている。この FORTRAN にはシステムの組込み関数・ライブラリ関数が用意され使うことができたが、プログラムの構造化への基盤となる副プログラム(サブ・プログラム)の概念はなかった。その機能のために“sub-routine”, “function”, “call”, “return”, 各文が加えられ、また主プログラム(メイン・プログラム) / 副プログラム相互間でデータのアクセスを可能にするため、“common” 文を配列の拡張として取り入れた言語が 1958 年に発表された FORTRAN II である。FORTRAN は自動プログラミング言語の歴史を拓いたシステムであり、時代に即応して次の年表に示すように拡張が行われ現在に至っている。

- FORTRAN II (1958年), FORTRAN IV (1962年), FORTRAN 66 (1966年), FORTRAN 77 (1978年), Fortran 90 (1992年)

データ型の“倍精度実数型”, “複素数型”にみられるように FORTRAN は科学技術計算用プログラミング言語として位置づけられアルゴリズムの記述を中心に発展してきた。

Fortran 90 では時代の要請に応じて構造体の宣言機能, データ型にポインタ属性を追加しデータの記述機能の拡充を行った。

次に、プログラミング言語の歴史で他の言語に最も大きな影響をあたえ、言語の文法・規約のみならず言語そのものの記述法に大きな変革をもたらしたのは“ALGOL 60”<sup>(1.3)</sup>である。ALGOL 60 はアルゴリズムを記述するためにつくられたもので 1960 年に IFIP<sup>(1.4)</sup>によって制定された記述言語である。

その構文記述にバックス記法(BNF記法)が用いられ言語文法, 仕様が論理的且つ明確に規定された。ALGOL 60 はその後継言語として制定された“ALGOL 68”とともに、現在汎用プログラミング言語として普及している

---

(1.3) ALGOL 60: Revised Report on the algorithmic language ALGOL 60. Communication of the ACM, 1962.

(1.4) IFIP: International Federation for Information Processing の略称。

“PL/I”, “ADA”, “PASCAL”, “C”, “Objective-C”, “C++”, 更には, オブジェクト指向言語の基を拓いた “Simula 67”, その流れのなかに位置する “Smalltalk” 等の言語に大きな影響を与えた。

プログラミング言語の発展をその系譜で捉えると, 上に記した FORTRAN, ALGOL, あるいは, 事務処理・事務管理向きのデータ処理機能を備えた COBOL等の思想を受け継いで生まれた情報処理の汎用的記述を目指した言語群と, MIT の J.McCarthy の開発した LISP, Alain Colmerauer らによって開発された PROLOG 等, リスト処理やホーン節に基づく述語論理をベースにして開発された論理型, あるいは, 人口知能記述型の言語群に分けることができる。これらの言語群のうち Smalltalk を見てみると ALGOL/Simula の流れの中でオブジェクト指向をより特徴づけた言語であるが, LISP 系列に位置づけられる Loops, Flavor 等の言語に影響を与えている。FORTRAN, ALGOL のようにプログラミングのパラダイムに革新的な変革をもたらした言語を含めてプログラム記述言語は既開発のシステムの特徴を継承しつつ発展してきた。

FORTRAN が 1956年に開発されて以来プログラミング言語 (コンパイラ) は, 座席予約・金融業務にみられるサービス向上のためのオンライン・リアルタイム・システム化, 分散且つ広域にわたる業務の統合システム化のために要請されるネットワークへの対応, ルーチンワークの自動化, 判断・学習等人間の知的行動の実現を目指した AI 機能の導入等のニーズに応じて研究・開発が進められ現在に至っている。

この 40年のプログラミング言語の発展史を通して現在最も注目されているのが, データの抽象化をより進めてオブジェクト指向の機能を取り入れた記述言語の開発, あるいは, 既成言語にオブジェクト指向機能を追加したシステム全体を見直してエンハンスした言語である。前者の代表は, 1980 年代初期に開発された Smalltalk-80 が上げられ, 後者には, Objective-C, C++ が

上げられる。

次節以降に、汎用プログラミング言語の歴史を拓いた FORTRAN, および、データ構造の枠組みの中にアルゴリズムをとり入れ、オブジェクト指向の機能を含めて抽象データ型の記述を目指した“C++”をとりあげ、データ構造を中心にその形を構文記法 (BNF) を用いて記述しデータ体系の把握と情報分析システムとの関連に触れていくことにしたい。

### 1. 2. 2 FORTRAN の構文について

〈宣言文〉 ::= 〈DIMENSION文〉 | 〈EQUIVALENCE文〉 |  
          〈COMMON文〉 | 〈IMPLICIT〉 |  
          〈PARAMETER文〉 | 〈EXTERNAL文〉 |  
          〈INTRINSIC文〉 | 〈型宣言文〉 | 〈SAVE文〉

#### 《注記》

汎用プログラミング言語で最も古い歴史を持つ FORTRAN は、Fortran 90 が制定されるまでは、データ宣言文種、データ型を指定する〈型宣言文〉でみると、初期の FORTRAN と比較して〈データ型〉に DOUBLE PRECISION, COMPLEX, LOGICAL, CHARACTER等が加えられ、データ文 (初期設定), BLOCK DATA 文 (初期設定副プログラム) の新規命令が追加されているものの基本的には大きな機能拡張はされていない。

データ型の宣言・定義機能は、サブプログラムで疑似的に新しいデータ型を作り上げることは出来ても FORTRAN 自身の持つ文を用いて独自のデータ型を生成する機能を持たない。

〈DIMENSION文〉 ::= DIMENSION 〈配列宣言子〉 {, 〈配列宣言子〉 }

## 《注記》

配列の割当サイズは、寸法宣言子で指定され静的に決まり PL/I, C 等の言語にみられる動的割当を行うことはできない。

DIMENSION文によって宣言されるデータの型は、暗黙の型宣言 (IMPLICIT文) で決まる。その指定が無い場合は名前の第1文字で整数型か実数型かが決まる。

〈配列宣言子〉 ::= 〈配列名〉 ( 〈寸法宣言子〉 {, 〈寸法宣言子〉} )

〈配列名〉 ::= 〈英字名〉

〈寸法宣言子〉 ::= [ 〈寸法の下限〉 :] 〈寸法の上限〉

## 《注記》

配列宣言子は、“定配列宣言子”，“整合配列宣言子”，“疑寸法配列宣言子”に分けられ寸法指定子に許される指定が、定配列宣言子の場合には整数式のみで、整合配列宣言子の場合には整数も書くことが可能で、この宣言子を引用した時点で寸法が定まる。疑寸法配列宣言子の場合には最終次元の寸法の上限を“\*”で示す配列宣言で宣言される。この配列は仮引数にのみ用いられる。

〈EQUIVALENCE文〉 ::= ( ( 〈変数名〉 | 〈配列要素名〉 |  
 〈配列名〉 | 〈部分列名〉 ) )  
 { ( ( 〈変数名〉 | 〈配列要素名〉 |  
 〈配列名〉 | 〈部分列名〉 ) ) }

## 《注記》

EQUIVALENCE文は同じ記憶域を二つ以上の要素で共有させる場合に用いられる。PL/I の DEFINE文はこの考え方を継いでいる。

変数名、配列要素名、配列名、部分列名は英字名で表される。

〈COMMON文〉 ::= COMMON [ / [ 〈共通ブロック名〉 ] / ]  
 ( 〈変数名〉 | 〈配列名〉 | 〈配列宣言子〉 )

{, [ / [ 〈共通ブロック名〉 ] / ]  
 ( 〈変数名〉 | 〈配列名〉 | 〈配列宣言子〉 ) }

## 《注記》

FORTRAN IIで SUBROUTINE, FUNCTION文が追加されサブプログラムの枠を越えた広域変数の定義をCOMMON文で行えるようにした。

〈IMPLICIT文〉 ::= IMPLICIT 〈データ型〉 ( ( 〈英字名 | 英字範囲〉 )  
 {, ( 〈英字名 | 英字範囲〉 ) } )  
 {, 〈データ型〉 ( ( 〈英字名 | 英字範囲〉 ) )  
 {, ( 〈英字名 | 英字範囲〉 ) } ) }

〈英字範囲〉 ::= 〈英字名〉 - 〈英字名〉

〈PARAMETER文〉 ::= PARAMETER ( 〈英字名〉 = 〈定数式〉  
 {, 〈英字名〉 = 〈定数式〉 } )

〈初期値設定副プログラム〉 ::= BLOCK DATA 〈英字名〉  
 [ 〈IMPLICIT 文〉 | 〈型宣言文〉 |  
 〈COMMON文〉 | 〈DIMENSION文〉 |  
 〈DATA文〉 | 〈PARAMETER文〉 |  
 〈EQUIVALENCE文〉 | 〈SAVE〉 ]  
 END

〈型宣言文〉 ::= 〈データ型〉  
 [ 〈定数〉 | 〈変数〉 | 〈配列〉 | 〈文関数〉 |  
 〈外部関数〉 | 〈組込み関数〉 | 〈仮手続〉 |  
 〈配列宣言〉 ] [ \* 長さ指定 ]

〈データ型〉 ::= INTEGER | REAL | DOUBLE PRECISION | COM-  
 PLEX | LOGICAL | CHARACTER [ \* 長さ指定 ]

(A) 文字型以外の型宣言文



以下，省略。

### 《注記》

PL/I のデータ宣言部分を下記に示す。

〈宣言〉 ::= (DECLARE | DCL) 〈宣言句リスト〉 ;

〈宣言句リスト〉 ::= 〈宣言句〉 {, 〈宣言句リスト〉}

〈宣言句〉 ::= [ 〈レベル〉 ] 〈識別子〉 [ ( 〈次元属性リスト〉 ) ]  
[ 〈属性リスト〉 ] [SYSTEM]

〈レベル〉 ::= 〈10進整数〉

〈次元属性リスト〉 ::= 〈次元属性〉 {, 〈次元属性リスト〉}

〈次元属性〉 ::= [ 〈下限〉 : ] 〈上限〉 | \*

〈下限・上限〉 ::= 〈10進整数〉 | 〈式〉

〈属性〉 ::= 〈データ型属性〉 | 〈記憶域類属性〉 | 〈有効範囲属性〉 |  
〈整列属性〉

〈属性リスト〉 ::= 〈属性〉 | 〈属性リスト〉 〈空白リスト〉 〈属性〉

〈データ型属性〉 ::=

〈算術属性〉 | 〈列属性〉 | 〈ピクチャ属性〉 | 〈LIKE 属性〉 |

〈ラベル属性〉 | 〈ファイル属性〉 | 〈入口属性〉 | 〈領域属性〉 |

〈事象属性〉 | 〈タスク属性〉 | 〈組込入口属性〉 | 〈条件属性〉 |

〈総称名属性〉 | 〈ロケータ属性〉 |

### (1.5) 参考資料:

- (1) 「プログラミング言語C」第2版, B.W.カーニハン / D.M.リッチー著, 石田晴久訳, 共立出版, 1989.
- (2) 「ANSI Cプログラミング」, 西田親生著, 付録2 C言語の構文フロー, 啓学出版, 1991年.
- (3) 「TURBO C++」, PROGRAMMER'S GUIDE, BORLAND.
- (4) 「3050 HI-UX C++」, HITAC マニュアル, マニュアル番号 "3050-3-301-10", 1992.

### 1.2.3 C++言語の構文<sup>(1,5)</sup>

現在パーソナル・コンピュータから超大型コンピュータまでの幅広いレンジで用いられている汎用プログラミング言語といえば第一にC言語をあげることができる。

さらに、近未来の情報処理記述言語として注目を集めているのがC言語にオブジェクト指向の機能を取り入れた“C++”言語である。

前節で触れたFORTRANとC++の新旧2つの代表的な汎用プログラミング言語の機能を比較対照することによって同記述言語の発展経緯，言語文法の特徴，新情報処理環境が要請する記述言語機能，等々を読み取ることができる。

#### 1.2.3 a データ構造部分

《記》 行の頭の“#”記号は，C言語に付加されたC++言語の新機能。

\*\* 宣言

〈宣言〉 ::= [ 〈宣言指定子〉 ] [ 〈宣言指定子リスト〉 ] |

〈アセンブラ宣言〉 | 〈関数宣言〉 | 〈結合宣言〉

〈宣言指定子〉 ::= 〈記憶クラス指定子〉 | 〈型指定子〉 |

# 〈関数指定子〉 |

〈テンプレート指定子〉 |

# friend |

typedef

〈宣言指定子リスト〉 ::= [ 〈宣言指定子リスト〉 ] |

〈宣言指定子〉

〈記憶クラス指定子〉 ::= auto | register | static | extern

<型指定子> ::= <単純型名> |  
                   <クラス指定子> | <列挙指定子> |  
                   <厳密型指定子> | const | volatile  
 # <関数指定子> ::= inline | virtual  
 <単純型名> ::= <クラス名> | typedef名 | char | short | int |  
                   long | signed | unsigned | float | double | void  
 <厳密指定子> ::=  
                   <クラス・キーワード> ( <識別子> | <クラス名> ) |  
                   enum <列挙型名>  
 <クラス・キーワード> ::= class | struct | union  
 <列挙指定子> ::= enum [ <識別子> ] { <列挙リスト> }  
 <列挙リスト> ::= <列挙子> | <列挙リスト> , <列挙子>  
 <列挙子> ::= <識別子> | <識別子> = <定数式>  
 <定数式> ::= <条件式>  
 <条件式> ::= <論理OR式> |  
                   <論理OR式> ? <式> : <条件式>  
 # <結合宣言> ::= extern <文字列> { <宣言リスト> } |  
                   extern <文字列> <宣言>  
 <宣言リスト> ::= <宣言> | <宣言リスト> ; <宣言>  
  
 \* \* 宣言子  
 <宣言子リスト> ::= <初期化宣言子> |  
                   <宣言子リスト> , <初期化宣言子>  
 <初期化宣言子> ::= <宣言子> [ <初期化子> ]  
 <宣言子> ::= <宣言名> | <ポインタ演算子> <宣言子> |  
                   <宣言子> ( <引数宣言子リスト> ) [ <変換限定子リスト> ] |

```

    <宣言子> [ [定数式] ] | ( <宣言子> )
<ポインタ演算子> ::= * [ <変換限定子リスト> ] |
    & [ <変換限定子リスト> ] |
    <クラス名> :: * [ <変換限定子リスト> ]
# <変換限定子リスト> ::= <変換限定子> [ <変換限定子リスト> ]
<変換限定子> ::= const | volatile
<宣言名> ::= <名前> |
#         <クラス名> |
#         ~ <クラス名> |
         typedef名
<型名> ::= <型指定子> [ <抽象宣言子> ]
<抽象宣言子> ::= <ポインタ宣言子> [ <抽象宣言子> ] |
    [ <抽象宣言子> ] ( <引数宣言リスト> )
    [ <変換限定子リスト> ] |
    [ <抽象宣言子> ] [ [定数式] ] |
    ( <抽象宣言子> )
# <引数宣言リスト> ::= <引数宣言> |
    <引数宣言リスト> , <引数宣言>
<引数宣言> ::= <宣言指定子> <宣言子> |
#         <宣言指定子> <宣言子> = <式> |
    <宣言指定子> [ <抽象宣言子> ] |
#         <宣言指定子> [ <抽象宣言子> ] = <式>
<関数宣言> ::= [ <宣言指定子> ] <宣言子>
    [ <コンストラクタ初期化子> ] <関数本体>
<関数本体> ::= <複合文>
<初期化子> ::= = <式> |

```

```

                                = { <初期化子リスト> [, ] } |
#                                ( <式リスト> )
<初期化子リスト> ::= <式> |
                                <初期化子リスト> , <式> |
                                { <初期化子リスト> [, ] }

**クラス宣言子
# <クラス指定子> ::= <クラス・ヘッド> { <メンバ・リスト> }
<クラス・ヘッド> ::=
    <クラス・キーワード> [ <識別子> ] [ <基本指定> ] |
    <クラス・キーワード> <クラス名> [ <基本指定> ]
<メンバ・リスト> ::= <メンバ宣言> [ <メンバ・リスト> ] |
    <アクセス指定子> [ <メンバ・リスト> ]
<メンバ宣言> ::= [ <宣言指定子> ] [ <メンバ宣言子リスト> ] ; |
    <関数定義> [ ; ] | <限定名> ;
<メンバ宣言子リスト> ::= <メンバ宣言子> |
    <メンバ宣言子リスト> <メンバ宣言子>
<メンバ宣言子> ::= <宣言子> [ <単純指定子> ] |
    [ <識別子> ] : <定数式>
<単純指定子> ::= = 0
<基本指定> ::= : <基本リスト>
<基本リスト> ::= <基本クラス指定子> |
    <基本リスト> , <基本クラス指定子>
<基本クラス指定子> ::= <クラス名> |
    virtual [ <アクセス指定子> ] <クラス名>
    <アクセス指定子> [ virtual ] <クラス名>
<アクセス指定子> ::= private | protected | public

```



## \* \* 式

〈式リスト〉 ::= 〈代入式〉 | 〈式リスト〉, 〈代入式〉

〈代入式〉 ::= 〈条件式〉 | 〈単項式〉 〈代入演算子〉 〈代入式〉

〈代入演算子〉 ::=

= | \*= | /= | %= | += | -= | >>= | <<= |

&= | ^= | |=

〈条件式〉 ::= 〈論理OR式〉 | 《記》重複記述

〈論理OR式〉 ? 〈式〉 : 〈条件式〉

〈論理OR式〉 ::= 〈論理AND式〉 |

〈論理OR式〉 || 〈論理AND式〉

〈論理AND式〉 ::= 〈論理OR式〉 |

〈論理AND式〉 && 〈論理OR式〉

〈包含OR式〉 ::= 〈排他OR式〉 | 〈包含OR式〉 | 〈排他OR式〉

〈排他OR式〉 ::= 〈AND式〉 | 〈排他OR式〉 ^ 〈AND式〉

〈AND式〉 ::= 〈等価式〉 | 〈AND式〉 & 〈等価式〉

〈等価式〉 ::= 〈関係式〉 | 〈等価式〉 ( == | != ) 〈関係式〉

〈関係式〉 ::= 〈シフト式〉 |

〈関係式〉 ( < | > | <= | >= ) 〈シフト式〉

〈シフト式〉 ::= 〈加算式〉 | 〈シフト式〉 ( << | >> ) 〈加算式〉

〈加算式〉 ::= 〈乗算式〉 | 〈加算式〉 ( + | - ) 〈乗算式〉

〈乗算式〉 ::= 〈ポインタメンバ式〉 |

〈乗算式〉 ( \* | / | % ) 〈ポインタメンバ式〉

〈ポインタメンバ式〉 ::= 〈キャスト式〉 |

# 〈ポインタメンバ式〉 ( \* | - ) \* ) 〈キャスト式〉

〈キャスト式〉 ::= 〈単項式〉 | ( 〈型名〉 ) 〈キャスト式〉

〈単項式〉 ::= 〈後置式〉 | ( ++ | -- ) 〈単項式〉 |

```

    <単項演算子> <キャスト式> |
    sizeof <単項式> | sizeof ( <型名> ) |
#
    <割当て式> |
#
    <分離割当て式>
    <単項演算子> ::= & | * | + | - | ! | ~
# <割当て式> ::=
    [::] new [ <配置> ] <限定型名> [ <初期化子> ]
    [::] new [ <配置> ] ( <型名> ) [ <初期化子> ]
# <配置> ::= ( <式リスト> )
# <限定型名> ::= <型指定子> [ <限定宣言子> ]
# <限定宣言子> ::= <ポインタ演算子> [ <限定宣言子> ] |
    <限定宣言子> [ <式> ]
# <分離割当て式> ::= [::] delete <キャスト式> |
    [::] delete [ <式> ] <キャスト式>
<後置式> ::= <一次式> |
    <後置式> [ <式> ] | <後置式> ( [ <式リスト> ] ) |
#
    <単純型名> ( [ <式リスト> ] ) |
    <後置式> . <名称> | <後置式> -> <名称> |
    <後置式> ( ++ | -- )
<一次式> ::= <リテラル> |
#
    this |
#
    :: <識別子> |
#
    :: <演算子関数名> |
    ( <式> ) | <名前>
<リテラル> ::= <整数定数> |
    <浮動小数点定数> | <文字定数> | <文字列>

```

```

<名前> ::= <識別子> |
#           <演算子関数名> |
#           <変換関数名> |
#           <限定名>
# <限定名> ::= <クラス名> :: <識別子> |
               <クラス名> :: <演算子関数名> |
               <クラス名> :: <変換関数名> |
               <クラス名> :: <クラス名> |
               <クラス名> :: ~ <クラス名>

**文
<文> ::= <ラベル付文> |
         <式文> | <複合文> | <選択文> |
         <繰返し文> | <ジャンプ文> |
#       <宣言>
<ラベル付文> ::= <識別子> : <文> |
                case <定数式> : <文> |
                default : <文>
<式文> ::= [ <式> ] ;
<複合文> ::= { [ <宣言リスト> ] [ <文リスト> ] }
<宣言リスト> ::= <宣言> | <宣言リスト> <宣言>
<文リスト> ::= <文> | <文リスト> <文>
<選択文> ::= if ( <式> ) <文> |
              if ( <式> ) <文> else <文> |
              switch ( <式> ) <文>
<繰返し文> ::= while ( <式> ) <文> |

```

```

do <文> while <式> ; |
for ( <for初期化文> ; [ <式> ] ; [ <式> ] ) <文>
<for初期化文> ::= <式文> | <宣言文>
<ジャンプ文> ::= break ; | continue ; | return [ <式> ] ; |
goto <識別子> ;

```

### 1. 2. 3 b 共通部分

```

<トークン> ::= <識別子> | <キーワード> | <定数> |
              <文字列リテラル> | <演算子> | <区切り>
<識別子> ::= <非数字> { <非数字> | <数字> }
<非数字> ::= _ | a | b | c | ..... | z | A | B | C | ..... | Z
<数字> ::= 0 | 1 | 2 | ..... | 9
<キーワード> ::=
    auto      | double  | int       | struct   | break    |
    else      | long   | switch   | case     | enum     |
    register  | typedef | char     | extern   | return   |
    union     | const  | float    | short    | unsigned |
    continue  | for    | signed   | void     | default  |
    goto      | sizeof | volatile | do       | if       |
    static    | while
<定数> ::= <整数定数> |
           <浮動小数点定数> | <文字定数> | <列挙定数>
<文字列リテラル> ::= " <文字> { <文字> } "
<2項演算子> ::= + | - | * | / | %
<関係演算子> ::= > | >= | < | <=

```

〈等価演算子〉 ::= = | !=  
 〈区切り〉 ::= [ ] | ( ) | { } | \* | , | : | = | ; | … | # |  
 〈宣言〉 ::= 〈宣言指定子〉 |  
           〈宣言指定子〉 〈初期化宣言子リスト〉  
 〈宣言指定子〉 ::= 〈記憶クラス指定子〉 | 〈型指定子〉 |  
                   〈型修飾子〉 | 〈型指定子〉 〈宣言指定子〉 |  
                   〈型修飾子〉 〈宣言指定子〉  
 〈記憶クラス指定子〉 ::= auto | register | static | extern | typedef  
 〈型指定子〉 ::= void | char | short | int | long | float | double |  
                   signed | unsigned |  
                   〈構造体共用体指定子〉 | 〈列挙型指定子〉 | 〈型定義名〉  
 〈型修飾子〉 ::= const | volatile  
 〈構造体共用体指定子〉 ::= (struct | union) 〈識別子〉 |  
                           (struct | union) { 〈構造体宣言リスト〉 } |  
                           (struct | union) 〈識別子〉 { 構造体宣言リスト }  
 〈構造体宣言リスト〉 ::= 〈構造体宣言〉 |  
                           〈構造体宣言リスト〉 〈構造体宣言〉  
 〈構造体宣言〉 ::= 〈指定子修飾子リスト〉 〈構造体宣言子リスト〉  
 〈指定子修飾子リスト〉 ::= 〈型指定子〉 |  
                           〈型指定子〉 〈指定子修飾子リスト〉 |  
                           〈型修飾子〉 | 〈型修飾子〉 〈指定子修飾子リスト〉  
 〈構造体宣言子リスト〉 ::= 〈構造体宣言子〉 |  
                           〈構造体宣言子リスト〉, 〈構造体宣言子〉  
 〈構造体宣言子〉 ::= 〈宣言子〉 |  
                   : 〈定数式〉 | 〈宣言子〉 : 〈定数式〉  
 〈宣言子〉 ::= 〈ポインター〉 |

〈直接宣言子〉 | 〈ポインター〉 〈直接宣言子〉  
 〈直接宣言子〉 ::= 〈識別子〉 |  
     ( 〈宣言子〉 ) | 〈直接宣言子〉 [ ] |  
     〈直接宣言子〉 [ 〈定数式〉 ] |  
     〈直接宣言子〉 ( 〈仮引数型リスト〉 ) |  
     〈直接宣言子〉 ( ) | 〈直接宣言子〉 ( 〈識別子リスト〉 )  
 〈ポインター〉 ::= \* |  
     \* 〈型修飾子リスト〉 | \* 〈ポインター〉 |  
     \* 〈型修飾子リスト〉 〈ポインター〉  
 〈型修飾子リスト〉 ::= 〈型修飾子〉 |  
     〈型修飾子リスト〉 〈型修飾子〉  
 〈定数式〉 ::= 〈条件式〉  
 〈条件式〉 ::= 〈論理OR式〉 | 〈論理OR式〉 ? 〈式〉 : 〈条件式〉  
 〈論理OR式〉 ::= 〈論理AND式〉 |  
     〈論理OR式〉 || 〈論理AND式〉  
 〈論理AND式〉 ::= 〈包含OR式〉 |  
     〈論理AND式〉 && 〈包含OR式〉  
 〈包含OR式〉 ::= 〈排他OR式〉 | 〈包含OR式〉 | 〈排他OR式〉  
 〈排他OR式〉 ::= 〈AND式〉 | 〈排他OR式〉 ^ 〈AND式〉  
 〈AND式〉 ::= 〈等価式〉 | 〈AND式〉 & 〈等価式〉  
 〈等価式〉 ::= 〈関係式〉 | 〈等価式〉 == 〈関係式〉 |  
     〈等価式〉 != 〈関係式〉 )  
 関係式 ::= 〈シフト演算子〉 |  
     〈関係式〉 ( ( | ) | (= | ) = ) 〈シフト演算子〉  
 〈シフト演算子〉 ::= 〈加法演算子〉 |  
     〈シフト演算子〉 ( ( | ) ) 〈加法演算子〉

〈加法演算子〉 ::= 〈乗法演算子〉 |  
                   〈加法演算子〉 ( + | - ) 〈乗法演算子〉  
 〈乗算演算子〉 ::= 〈キャスト式〉 |  
                   〈乗法演算子〉 ( \* | / | % ) 〈キャスト式〉  
 〈キャスト式〉 ::= 〈単項演算子〉 |  
                   ( 〈型名〉 ) 〈キャスト式〉  
 〈単項演算式〉 ::= 〈後置式〉 | ( ++ | -- ) 〈単項演算式〉 |  
                   〈単項演算子〉 〈キャスト式〉 |  
                   sizeof 〈単項演算式〉 | sizeof ( 〈型名〉 )  
 〈単項演算子〉 ::= & | \* | + | - | ~ | !  
 〈後置式〉 ::= 〈1次式〉 |  
                   〈後置式〉 [ 〈式〉 ] | 〈後置式〉 ( ) |  
                   〈後置式〉 ( 〈仮引数式リスト〉 ) |  
                   〈後置式〉 . 〈識別子〉 | 〈後置式〉 -> 〈識別子〉 |  
                   〈後置式〉 ++ | 〈後置式〉 --  
 〈引数式リスト〉 ::= 〈代入式〉 | 〈引数式リスト〉 , 〈代入式〉  
 〈型名〉 ::= 〈指定子修飾子リスト〉 |  
                   〈指定子修飾子リスト〉 〈抽象宣言子〉  
 〈抽象宣言子〉 ::= 〈ポインター〉 | 〈直接抽象宣言子〉 |  
                   〈ポインター〉 〈直接抽象宣言子〉  
 〈直接抽象宣言子〉 ::= ( 〈抽象宣言子〉 ) |  
                   〈直接抽象宣言子〉 [ ] | [ 〈定数式〉 ] |  
                   〈直接抽象宣言子〉 [ 〈定数式〉 ] |  
                   〈直接抽象宣言子〉 ( 〈仮引数型リスト〉 )  
 〈仮引数型リスト〉 ::= 〈仮引数リスト〉 | 〈仮引数リスト〉 , ...  
 〈仮引数リスト〉 ::= 〈仮引数宣言〉 |

〈仮引数リスト〉, 〈仮引数宣言〉  
 〈仮引数宣言〉 ::= 〈宣言指定子〉 | 〈宣言指定子〉 〈宣言子〉 |  
                   〈宣言指定子〉 〈抽象宣言子〉  
 〈初期化宣言子リスト〉 ::= 〈初期化宣言子〉 |  
                                   〈初期化宣言子リスト〉, 〈初期化宣言子〉  
 〈初期化宣言子〉 ::= 〈宣言子〉 | 〈宣言子〉 = 〈初期値式〉  
 〈初期値式〉 ::= 〈代入式〉 |  
                   { 〈初期値式リスト〉 } | { 〈初期値式リスト〉 }, |  
 〈初期値式リスト〉 ::= 〈初期値式〉 |  
                                   〈初期値式リスト〉, 〈初期値式〉  
 〈代入式〉 ::= 〈条件式〉 |  
                   〈単項式〉 〈代入演算子〉 〈代入式〉  
 〈代入演算子〉 ::= = | \*= | /= | %= | += | -= | <(<= | >=) = | &= | ^= | |=  
 〈列挙型指定子〉 ::= 〈列挙識別子〉 |  
                   { 〈列挙子リスト〉 } |  
                   〈列挙識別子〉 { 〈列挙子リスト〉 }  
 〈列挙子リスト〉 ::= 〈列挙子〉 | 〈列挙子リスト〉 , 〈列挙子〉  
 〈列挙子〉 ::= 〈識別子〉 | 〈識別子〉 = 〈定数式〉  
 〈型定義名〉 ::= 〈識別子〉

### 1.3 SECRETARYの構文記述

#### 《記》

- 〈E〉は、エンター・キーを押す操作を表すものとする。
- 〈E〉のみの場合は、直前のメッセージをコマンドラインに呼び出す。
- { } は、その括弧内の形が0回以上繰り返されることを表すものとする。

る。

- [ ] は、オプション指定を表す。

〈メッセージ〉 ::= { 〈メッセージ単位〉 } 〈E〉

〈メッセージ単位〉 ::= 〈コマンド集合〉 [ 〈カタログ名リスト〉 ]

〈コマンド集合〉 ::= 〈コマンド〉 { : 〈コマンド〉 }

〈コマンド〉 ::= 〈機能部〉 |

〈機能部〉 〈述部〉 |

〈複合文〉

〈カタログ名リスト〉 ::= 〈カタログ名〉 { : 〈カタログ名〉 }

〈複合文〉 ::=

DO 〈スカラー変数〉 = 〈初期値〉 TO 〈終値〉 [BY 〈増分値〉] :

〈コマンド集合〉 END: |

DO WHILE( 〈条件式〉 ): 〈コマンド集合〉 END:

《記》 初期値・終値・増分値は、整数。

《注記》

- コマンドの終端が “;” で終わる場合は当該コマンドの機能部が次のコマンドに引き継がれる。
- 〈メッセージ〉 が1ステートメントの場合はコマンド終端記号 ( “;” または “.” ) を省略することが出来る。
- メッセージ単位が空の場合は直前に入力したコマンドが画面に呼び出される。その情報に修正を加えて実行をすることも出来る。
- カタログの内容がコマンド集合の場合、カタログ名を書くことでカタログの内容が画面 (メッセージ・ストリング) に展開される。
- カタログにはコマンド集合の他、会社名リスト、項目名リスト等のデータも入れられる。

それらは、文によって引用の仕方とその働きが異なる。

---

〈機能部〉 ::=

AGK	AGM	AGS	CALL	CFILE	
CHANGECODE	CC	CHANGEDATE	CD	CHANGETITLE	
CT	CLEARACTIVE	CA	COLUMN	COL	
COMPUTE	COM	DATA	T	DATACROSS	
DTC	DATAG	DTG	DATAMPL	DTM	
DBLIST	DBL	DELETE	DEL	DISPLAY	
D	DMX	DMY	DO	DTCPLOT	
END	ESTIMATE	EST	ESTY	ESTYS	
FIELD	F	FILE	GET	G	
GETBASE	GB	GETCRS	GETC	GETPUTC	
GETP	GETREPEAT	GETR	GETUSER	GU	
HARDCOPY	HC	HELP	H	HIST	
IF	INFO	I	INFOBASE	IB	
INFOUSER	IU	LET	LINE	LIST	
L	LISTOPTION	LOP	LSM	LSMS	
LSMC	LSMCS	LSMN	LSMNS	MATRIX	
MAT	MODE	MOVINGAV	MAV	MXLIST	
ML	MXUPDATE	MU	NOPAGE	OPEN	
OUTDEV	OD	OUTHC	OUTH	PAGE	
PG	PERIOD	PD	PLOT	PL	
PUTBASE	PB	PUTCATALOG	PC	PUTCRS	
PUTC	PUTUSER	PU	RANDEXP	RNE	
RANDNOR	RNN	RANDPOIS	RNP	RANDUNIF	
RNU	READMT	RDM	READSUM	RDS	
READTMPF	RDT	RENAME	REN	SAVE	
S	SCATTER	SC	SCATTERN	SCN	
SCATTERS	SCS	SKIP	SORT	STATIS	
ST	SUMMATON	SUM	TITLE	T	
TRANSPOSE	TR	TREND	TRD	UPDATE	
U	UPDATECAT	UC	OVER	/*	

---

## 《記》

機能（コマンド）名の長いものについては、その省略形が定められている（省略形はフルネームの直後に入れられているためアルファベット順にはなっていない）。

〈述部〉 ::= 〈算術代入文〉 |  
           〈句リスト〉 [ 〈条件式〉 ] |  
           [ 〈句リスト〉 ] 〈条件式〉

〈算術代入文〉 ::= 〈算術変数〉 = 〈算術演算式〉 |  
                   〈行列変数〉 = 〈行列演算式〉

〈算術変数〉 ::= 〈スカラー変数〉 |  
                   〈ベクトル変数〉 |  
                   〈添字付ベクトル変数〉

〈スカラー変数〉 ::= @〈名前〉

〈ベクトル変数〉 ::= 〈名前〉

〈添字付ベクトル変数〉 ::= 〈名前〉 ( 〈添字式リスト〉 )

〈添字式リスト〉 ::= 〈添字式〉 |  
                   [ 〈添字式リスト〉 ] , 〈添字式〉

## 《注記》

添字式は位置パラメータで3引数返書くことが可能で、それぞれ次のように定義される。

- 1 番目引数：1次元配列の要素，または区間指定時の開始要素
- 2 番目引数：区間指定時の終端要素
- 3 番目引数：1次元配列全体の変位

## 《添字付ベクトル変数の例》

VECT1(2) …………… 1次元配列 VECT1 の2番目の要素

VECT1(@K) ..... 1次元配列 VECT1 の @K 番目の要素  
 (@K はスカラー変数)

VECT1(@K+5) ..... 1次元配列 VECT1 の @K+5 番目の要素

VECT1(3,5) ..... 1次元配列 VECT1 の 3～5 番目迄の要素

VECT1(,,-1) ..... 1次元配列 VECT1 の左への1要素変位  
 (時系列データの場合は1期ラグを表す)

〈添字式〉 ::= 〈定数〉 | 〈スカラー変数〉 |  
 〈スカラー変数〉 〈演算子〉 〈定数〉

《記》添字式に限り演算子 “\* \* ” 記号を除く。

〈算術演算式〉 ::=  
 [+ | - ] 〈算術演算項〉 { 〈演算子〉 〈算術演算項〉 } |  
 [+ | - ] [ ( ) 〈算術演算式〉 () ]  
 { 〈演算子〉 [ ( ) 〈算術演算式〉 () ] }

〈算術演算項〉 ::= 〈定数〉 | 〈算術変数〉 | 〈算術関数引用〉

〈算術関数引用〉 ::= 〈算術関数名〉 ( 〈算術変数〉 )

〈算術関数名〉 ::=

ABS		COS		CUMU		EPA		EXP	
EXPECT		LOG		LOG10		MEAN		NOWB	
NOWD		SDEV		SIN		SQRT		TAN	
TBLN		TOTAL							

《算術関数の説明》

ABS(V) ..... 絶対値

COS(V) ..... 余弦

CUMU(V) ..... 累積値

EPA(V) ..... 季節調整 (EPA法)

EXP(V) ..... 指数

EXPECT(V,P).....	期待値 (V:データ, P:確率)
LOG(V) .....	自然対数
LOG10(V) .....	常用対数
MEAN(V).....	平均値 (スカラー)
NOWB(V).....	レコード内語数 (ベースエリア)
NOWD(V).....	レコード内語数 (カタログファイル)
SDEV(V) .....	標準偏差
SIN(V) .....	正弦
SQRT(V).....	平方根
TAN(V) .....	正接
TBLN(S).....	正規分布累積確率
TOTAL(V) .....	合計 (スカラー)

《記》 V : ベクトル, P・S : スカラー。

〈演算子〉 ::= + | - | \* | / | \*\*

〈行列変数〉 ::= 〈名前〉 | 〈#表示子〉 |  
 〈添字付行列変数〉

#### 《注記》

行列変数には、行列属性を持つもの、ベクトル属性を持つもの、スカラー属性を持つものに分けられる。それがどの属性の変数であるかは、システムへの登録時に使用者がその何れかの属性を明示的に指定することで決められる場合と、右辺の行列演算の結果から左辺の変数の属性をシステムが自動的に判別して登録する場合とに分けられる。

また、次の表記例のように行列属性を持つ行列変数は添字指定によってベクトル、或いはスカラーを表し、ベクトル属性を持つ行列変数ならばその指定によってスカラーを表すことが出来る。

#### 《行列変数の表示例》

MX	……………	行列 MX
MX(2,4)	……………	行列 MX の 2～4 行, 全列の小行列
MX(/3,8)	……………	行列 MX の全行, 3～8 列の小行列
MX(2,4/3,8)	……………	行列 MX の 2～4 行, 3～8 列の小行列
MX(4)	……………	行列 MX の 4 行目の行ベクトル
MX(/8)	……………	行列 MX の 8 列目の列ベクトル
MX(2/3,8)	……………	行列 MX の 2 行目, 3～8 列の列ベクトル
VECT(5)	……………	ベクトル VECT の 5 番目の要素
VECT(2,7)	……………	ベクトル VECT の 2～7 番目迄の要素
MX(3/8)	……………	行列 MX の 3 行, 8 列の要素
MX(@I/@J)	……………	行列 MX の @I 行, @J 列の要素

〈添字付行列変数〉 ::= 〈名前〉 ( 〈行列配列記述子〉 ) |  
 〈#表示子〉 ( 〈行列配列記述子〉 )

〈#表示子〉 ::= # 〈名前〉

#### 《注記》

#表示は, 2次元の配列平面を持ちシステムの動作用記憶域 (SECRETARY ではアクティブ・エリア) を表す代名詞として用いられる。

データベースを含めデータ検索で得られたレコードはこの記憶域に入れられる。

また, #表示の添字の書き方は行列変数に準じる。

〈行列配列記述子〉 ::= 〈配列限界・行〉 | / 〈配列限界・列〉 |  
 〈配列限界・行〉 / 〈配列限界・列〉

〈配列限界〉 ::= 〈添字式〉 | {, 〈添字式〉 }

〈行列演算式〉 ::=  
 (+ | -) 〈行列指定子〉 { 〈演算子〉 〈行列指定子〉 } |  
 (+ | -) [( ) 〈行列演算式〉 () ]

{ <演算子> [( ) <行列演算式> 0 ] }

《注記》

行列演算における演算子の定義について

“\*”記号は行列積，他の演算子是对应する要素間の二項演算。

対応する要素間の乗算は関数 (MULT) による。

<行列指定子> ::= <行列変数> | <行列関数> |  
 <小行列関数> | <転置行列表示子> |  
 <逆行列表示子>

<行列関数> ::= <行列関数名> ( <行列変数> )

<行列関数名> ::=

ABS		CORX		COS		COVX		CUMU	
DET		DEVX		DIAG		EIGNH		EIGNQ	
EXP		IDENT		LOG		LOG10		MEAN	
MULT		NORD		PRODX		SDEV		SIN	
SIZE		SQRT		SUBMX		SUM		SUMX	
SUMY		TAN		TOTAL		TRACE			

《行列関数の説明》

ABS(X) ..... 行列 X の絶対値  
 CORX(X) ..... 行列 X の相関行列  
 COS(X) ..... 行列 X の余弦  
 COVX(X) ..... 行列 X の分散共分散行列  
 CUMU(X) ..... 行列 X の累積値  
 DET(X) ..... 行列 X の行列式 (スカラー)  
 DEVX(X) ..... 行列 X の偏差積和行列  
 DIAG(V) ..... 行ベクトル V を対角行列に変換  
 EIGNH(X) ..... 固有値および固有ベクトル (実対称行列)

EIGNQ(X) …… 固有値および固有ベクトル (実非対称行列)  
 EXP(X) …… 行列 X の指数  
 IDENT(N) …… 単位行列 (N は行列のサイズ)  
 LOG(X) …… 行列 X の自然対数  
 LOG10(X) …… 行列 X の自然対数  
 MEAN(X) …… 行列 X の平均値 (出力: ベクトル)  
 MULT(X,Y) …… 行列 X と Y の対応する要素の積  
 NORD(X) …… 標準化データ行列  
 PRODX(X) …… 行列 X の積和行列  
 SDEV(X) …… 行列 X の標準偏差  
 SIZE(X) …… 行列 X のサイズ [出力: ベクトル (1×2)]  
 SQRT(X) …… 行列 X の平方根  
 SUBMX(i) …… 小 (部分) 行列関数  
 SUM(X) …… 行列 X の全要素和 (出力: スカラー)  
 SUMX(X) …… 行列 X の行和 (出力: ベクトル)  
 SUMY(X) …… 行列 X の列和 (出力: ベクトル)  
 TAN(X) …… 行列 X の正接  
 TOTAL(X) …… 合計値 (出力: ベクトル)  
 TRACE(X) …… 行列 X のトレース (対角要素の和)

〈小行列関数〉 ::=

SUBMX ( 〈小行列識別子〉 { // 〈小行列識別子〉 } ) |

SUBMX ( 〈小行列識別子〉 { %% 〈小行列識別子〉 } )

《注記》

“//” 記号は列による結合, “%%” は行による結合を表す。

〈小行列識別子〉 ::=

〈名前〉 / 〈小行列記述子〉 / 〈小行列記述子〉 / |  
 〈#表示子〉 / 〈小行列記述子〉 / 〈小行列記述子〉 /

《注記》

第2パラメターの小行列記述子は行指定, 第3のそれは列指定を行う

〈小行列記述子〉 ::= 〈自然数〉 { , [ - ] 〈自然数〉 }

《注記》

“-”記号は区間指定を行う特殊記号。

例えば, 1, -5 は1行(列)から5行(列)を表す。

〈転置行列表示子〉 ::= 〈行列変数〉’

〈逆行列表示子〉 ::= 〈行列変数〉”

〈句リスト〉 ::= 〈句〉 |

〈句リスト〉 〈句〉

〈句〉 ::= [ 〈キーワード〉 ] 〈システム識別子〉 |

〈算術変数〉 | 〈行列変数〉

〈システム識別子〉 ::= 〈名前〉 |

〈定数〉 | 〈システムファイル名〉 | 〈カタログ〉 |

〈データベース名〉 | 〈フィールド名〉 | 〈索引名〉 |

〈拡張項目コード〉 | 〈項目コード〉

〈システムファイル名〉 ::= USER | BASE | TMPF

〈カタログ〉 ::= 〈文字列集合〉

《注記》

文字列集合には, コマンド集合, 会社コード・項目コードその他のテキスト型情報を含む。

カタログ名の頭に“\$”記号がつけられた名前がコマンドのオペランドに書かれた場合処理系は特別の識別子として認識する。

例えば、データベース、システムファイル等のレコードの検索系コマンドに用いられた時はその名前のレコードに入れられている会社コード・項目コード等を展開しコマンドの情報として取り込む。

- “¥¥名前” : ユーザが自由に登録・利用できる名前。
- “¥¥¥名前” : 使用対象は全ての利用者であるが、この名前を用いたカタログの登録はシステム管理者のみに権限が与えられる。

#### 《使用例》

- (1) 分析対象の企業グループの会社コードがカタログ名 “¥¥COGROUPOU1” に、選択する複数の財務項目が “¥¥ACCITEM1” に登録されている場合、その対象レコード全てを検索するコマンドを次に示す。

```
GETR ¥¥COGROUPOU1 REPEAT ¥¥ACCITEM1:
```

- (2) EXSTATデータベース検索用に登録されているシステムカタログを用いてレコードを検索する例を次に示す。

“¥¥¥EXSIC18” は、建設業（18はEXSTATの業種分類コード）の会社グループのシステムカタログ名。

```
GETR ¥¥¥EXSIC18 REPEAT ¥¥ACCITEM1
¥¥ACCITEM2:
```

〈拡張項目コード〉 ::=

```
〈システムファイル名〉 . 〈項目コード〉 |
〈データベース名〉 . 〈項目コード〉
```

〈条件式〉 ::= 〈条件句〉 | { 〈接続子〉 〈条件句〉 }

〈条件句〉 ::=

```
〈欄名〉 〈関係子〉 〈定数リスト〉 |
〈欄名〉 〈関係子〉 〈スカラー変数〉
```

〈欄名〉 ::= 〈索引無欄名〉 | 〈索引付欄名〉

〈接続子〉 ::= & | ! | ^

〈関係子〉 ::= = | ^= | > | < | >= | <= | ~

〈キーワード〉 ::=

ADD	ALL	AND	BASIC	BASIS	
BY	CALL	CONTROL	DATA	DECIMAL	
DEL	DISPLAY	DO	EDIT	END	
FIELD	FOR	FORM	FROM	GET	
INPUT	INTO	KEY	LINE	LIST	
LOAD	LP	LPRINT	MASK	MODEL	
NOT	ON	OR	OUTPUT	PAGE	
PERIOD	PUT	READ	SAVE	SKIP	
SORT	SPAN	THEN	TO	VALUE	
WHERE	WHILE	WHR	WITH	WITHOUT	
ZERO					

〈名前〉 ::= [ 〈名前記号〉 ] 〈英字〉 { 〈英数字〉 }

〈名前記号〉 ::= \_ | . | # | @

《注記》

“ $\$$ ”, “#”, “@” の何れかの記号で始まる名前は、システムで特別に定義づけられるため使用上の制約がある。

- $\$$  で始まる名前は、データ検索系コマンドで特別の扱いをする。
- # で始まる名前は、動作記憶域 (ACTIVE) を指す代名詞として用いられる。
- @ で始まる名前は、スカラー変数として用いられる。

〈英数字〉 ::= 〈英字〉 | 〈数字〉

〈リテラル〉 ::= 〈数値定数〉 | 〈文字定数〉  
 〈数値定数〉 ::= [ + | - ] 〈符号無し整数〉 |  
                   [ + | - ] 〈符号無し実数〉  
 〈文字定数〉 ::= '文字列'

#### 1.4 プログラミング言語と情報分析システムの構文記述について

プログラミング言語 (programming language) は、人間が問題を作ったり解いたりするときに抱く概念・アルゴリズム等を、形式化した論理で記述する言語のことをいい、利用目的・利用分野・言語水準・言語構造等によって種々の分け方がされている。

それを対照法的に表してみると、

##### (1) 言語処理 (記述) 系とシステム制御系 (記述)

言語処理系は、FORTRAN, COBOL, Pascal等。システム制御系は、アセンブラーがそれを代表する。C, PL/Iは、現在言語処理用として用いられる場合が多いが中間に位置しシステムを制御する機能を持っている (C言語は UNIX の記述言語で、PL/I は Multics の記述言語)。

##### (2) 汎用言語と問題向き言語

前者は、一般的な問題を記述できる万能型言語、後者は、特殊な分野の問題を能率良く記述出来る言語。

##### (3) 手続き型と非手続き型言語

手続き型は、記述を順序立て明示的に与える方式で汎用プログラミング言語はこれに属する。非手続き型は、名前の通りでデータの定義、またデータ相互の関係・データと処理との関係等をランダムに記述する方式で、論理型言語のProlog、表形式言語 (RPG等) 等はこれに属する。

##### (4) 高水準と低水準言語

人間とコンピュータを対置し、人の論理に近い形で記述できるものを高水準言語と云い、コンピュータに近いものを低水準言語と云っている。

この章で取り上げたFORTRANは、汎用プログラミング言語の最初のものであり同言語によって“配列”というデータ構造の概念が生まれ、“EQUIVALENCE”文によって物理的に表される記憶装置の位置と言語上で表される論理的な名前の関係を結びつける働きを持たせた。

配列は、データ構造の最も基本的な型であり現有のプログラミング言語ではプログラム実行時に割り当てる機能、調整可能な広がりを持つ配列定義、多次元配列の特定の次元の断面を総称的に表すこと等へと発展してきている。

EQUIVALENCE文は、複数のデータ形式として宣言した変数が同一の記憶域に割り付けられる。この働きを利用することによりシステム記述にアプローチ出来るためシステム開発者の人気を集めた。この文の概念は、PL/I言語のDEFINE文、C言語のUNION文等へ受け継がれている。

FORTRAN は、プログラミング言語の元祖でありながら“FORTRAN IV”，“FORTRAN 66”，“FORTRAN 77”と文法を改訂し、この章で記述した“FORTRAN 90”へと発展した言語である。市民権を得たプログラミング言語は、それで記述したプログラム資産の蓄積と共にそのプログラム市場が拡大し、言語機能の拡充・発展を促していくことを示した言語でもある。

FORTRAN は、手続き型で言語処理系の高水準言語であるが EQUIVALENCE文の機能、通常組み込まれているbyte（およびbit）組込関数等の記憶域制御機能と、これらの処理に関連する機能を補強したサブ・プログ

---

(1.6) SECRETARY 開発に於いて参考にした情報分析システム“ASTRO-FOIL（開発代表者：元和歌山大学教授杉浦一平氏）”，“STEPS（開発代表者：元神戸大学教授定道 宏氏）”等はシステム記述言語にFORTRAN が採用されている。

ラムを拡充することによって情報分析システムを記述することが出来る。<sup>(1.6)</sup>

FORTRAN をシステム記述言語にすることによってFORTRANで書かれた数理解析の膨大なプログラム資産を情報分析システムに円滑に取り入れていくことが出来る。

C および C++言語は、現在ではアプリケーション・プログラムを記述する言語としてスーパー・コンピュータからパーソナル・コンピュータまでの広いレンジで用いられているが、本来は UNIX のシステム記述言語でもあるようにシステム記述言語として生まれ発展してきた言語である。

C (C++を含めて) 言語は、本章の構文記述 (1.2.3 項) で明らかなようにポインター (本稿 2.4.1項参照) の操作機能を強化することにより記憶域の制御を柔軟に行えるようにしていること、また、その機能を活用することにより複雑且つ巾広い構造体の宣言 (struct構造体指定子) を可能とし、データ処理のプログラム記述の中心をなすデータ構造の定義とその操作を円滑に行わしめている。

さらに、C++ 言語は従来の構造体機能を拡張しクラス (class キーワード) の概念を規定している。クラスの宣言では、構造体データのほかデータのアクセス制御・セキュリティー機能、導出クラス (継承) の定義機能を持たせること、クラス宣言の中にアルゴリズムの記述機能を持たせる等によって“オブジェクト指向プログラミング”に従ったプログラム作成基盤の提供を可能としている。

我々の開発した“SECRETARY”は、システム記述言語に PL/Iを採用したことにより、言語の特長であるファイルに関する記述機能と記憶域制御機能を情報分析システム (SECRETARY) に反映させることが出来た。

財務情報分析システム等の問題向きシステムの開発理念の第一は問題解決を最も能率的に行い得ることであり、したがって、目的言語の命令体系 (コマンド) は自ずと高水準で特殊化されたものになる。

しかし、分析システムの利用者は便宜性のみでなく各個人の分析ニーズを記述するためのマイクロな記述も含めた柔軟性を分析システムに求める。

我々は、この相反する課題に対して情報分析システム設計上次のような方法をとった。

- ・行列理論（演算）に従った表記法で行列演算を記述出来ること。
- ・分析用関数を無制限に容易に組み込めること。
- ・分析者固有のカタログ機能を充実し容易に引用出来るようにすること。

等である。

プログラミング言語は、FORTRAN が先覚者でありながら文法改訂を繰り返して現在の FORTRAN 90 を築き上げてきたように、利用からのニーズをオープンに受け入れることの出来るシステムをつくることである。長期間生き続ける情報システムは高い価値を持つシステムといえる。

これからの情報分析システム（SECRETARY）の課題は、分析者が持っている分析プログラムを分析者自身が情報分析システムに登録出来る手続きのツールをシステム機能として持っていること、C++ 言語が持っている“CLASS”機能をシステムに組み入れオブジェクト指向プログラムの作成・実行が出来るファシリティーをシステム機能として提供出来る仕組みを、情報分析システムが持つことである。



## 第2章 データ構造化の意義と新しい展開

### 2.1 データ構造化の意義と背景<sup>(2.1)</sup>

データとは、JIS用語によれば「人間または自動的手段によって行われる通信、解釈、処理に適するように形式化された事実、概念、指示の表現」と定義されている。

また、データ構造とは「データを記憶し操作するための表現形式」で、データ型を機械内部の表現として実現することである。<sup>(2.2)</sup>

データ構造は、データ処理の記述論理に関わる問題としてアルゴリズムと共に、またアルゴリズムと相互に深い関係を持つ概念で常に情報処理のシステム設計、プログラミング記述に関する研究の中心に位置する。

データ、およびデータ構造に関する問題領域は非常に広く研究・開発の取り組みも多分野にわたっており、現在までの理論と成果を完全に把握することは困難な状況にある。すなわち、データ構造の理論は基本的なデータの型に始まりその型、型の集合をコンピュータの内部表現の形で表すデータ構造、また、それらのデータに関する問題を記述するための概念とその操作体系であるデータモデル（階層モデル・網モデル・関係モデル等）、さらに最近注目を集めているデータ抽象化の問題等の概念的・理論的研究、データベース・システムへの応用等多岐を極める。

---

(2.1) 本項より2.4.2「抽象データ型と財務諸表」までは、下記拙稿を元に加筆・修正を行ったものである。

民野庄造, 「データ構造と財務諸表」, 経済経営研究叢書, 経営機械化シリーズ 21, 神戸大学経済経営研究所, 1993, pp.66-184.

(2.2) 長尾 真他編, 「情報科学辞典」, 岩波書店, 1990年, P.491.

プログラミングの手法で最近注目されているのが、“オブジェクト指向”のプログラミング・パラダイムに基づくプログラミングである。ここではデータ構造に関する記述機能をデータ処理の中でより重視した考えに基づいており、データとデータ処理のアルゴリズムを一体化した“抽象データ型”<sup>(4.3)</sup>を一層進めることによってますます複雑化していくプログラム構造に対処しようという概念である。

与えられた問題をコンピュータを用いて処理（システム化）するとき、問題を記述するためにコンピュータの持つ（提供する）情報処理言語、とりわけ、プログラミング言語に何を選択するかは重要な問題である。

最近ではプログラミングのモデルとして“関数型プログラミング”、“論理型プログラミング”、“オブジェクト指向プログラミング”が登場し従来からのプログラミング・スタイルである“手続き型（命令型）プログラミング”と合わせプログラムの設計の方法のモデルを提供する、“プログラミング・パラダイム”が注目されるようになった。

この新しい開発基盤の下に展開される“抽象データ型”に触れ、さらに財務諸表データに関して抽象データ型が如何に適用され得るかの可能性についても考察を進めていくことにする。

データ構造は、データを数値・文字・論理型等の型にクラス分けを行うこと、データの階層関係、他のデータとの関係などをなんらかの表現形式を用いてコンピュータの内部に実現することである。データ構造は、配列・リスト等に見られるように抽象的な概念として捉えられるという特徴を持っており、その構造のいかんは分析手続きや分析手法を記述すること、すなわちアルゴリズムと密接な関係を持ち処理能率などデータ処理全般に影響を与える。

単純な例を上げると、空白あるいはコンマで区切られた数値の並び、英数字

---

(4.3) 本稿, 2.4.1 「抽象データ型について」を参照。

・漢字などの文字列はデータとしては構造を持っていないデータである。これに対してFORTRAN等の汎用プログラミング言語で古くから使われてきている“配列”は最も単純な構造であるが構造化されているデータ構造の型である。

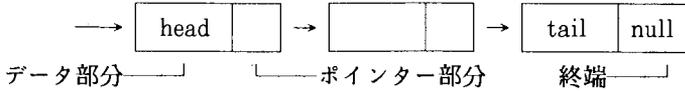
構造を持たないデータの場合は、プログラム中のデータ処理でデータをアクセスする度にデータ自身の属性に関する情報を確認しながら処理手続きを記述しておりデータと手続きが個別的なルールで関係づけられ、プログラムに求められる「一般的・慣用的表現」, 「プログラムの構造化」, 「プログラムの独立性」等を満たしていない。

後者の構造を持つデータの場合は、データ項目の属性に関する情報、項目間の関係等がデータ自身に記述されているためデータ構造に関する情報を参照して簡単に処理手続きが記述できる。データの構造化をより進めることによりデータと処理手続きの独立性を高め、アルゴリズムの要件を満たした規範的なプログラムを作成することが出来る。

## 2. 2 リスト構造について

リストは配列と並び文字列を表す基本的で且つデータ構造の代表的な型である。配列の場合は構造が単純であり記憶域が固定しているためデータ処理速度が早いという特徴を持つ。しかし記憶域が固定していることによりデータの挿入・削除等の更新処理は煩雑で処理速度も低下する。

配列に対してリストは配列の得失を裏返しにした特性を持つものと考えてよい。リストは次の図で示すように数値や文字列を入れる枠をつくり枠同志をポインターで接続（リンク）したものである。



## 《記》

“null (nilとも表される)” は、リストの終端で通常斜線で表され、“何も指示していない” という意味を表すのに用いる。PASCALでは“nil” が用いられる。

リスト構造は上に示したように1本の列より成る“線形リスト(linear-list)”，線形リストの終端の枠の前方ポインターを先頭の枠にリンクし環状形にした“環状リスト(circular-list)”，各枠に二つのポインターを設け隣の枠相互を結び付けた“双方向リスト(doubly-linked-list)”，同じく枠に二つのポインターを設け下位の枠にリンクしてつくる“2分木(binary-tree)”，データベースのデータ構造としてよく用いられている階層構造を構成する“木(tree)”，その外，言語プロセッサでよく用いられる“スタック(push-down stack)”，“待行列(queue)”等種々のものが確立されている。

このようにリスト構造はデータ構造のベースとなるもので、あらゆる分野のシステムのデータ構造記述に用いられている。現在ホットな研究開発のテーマである知識情報処理の分野でも欠くことの出来ない概念である。

### 2. 3 リスト処理による財務諸表データの記述

データの構造化は汎用プログラミング言語などの言語システムの持つ宣言文・定義文を用いて実現されるが、その標準機能のみでは記述が出来ないか冗長の記述によらなければならない場合もある。

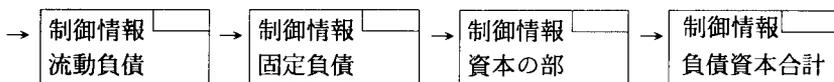
たとえば、財務諸表のうちの貸借対照表の上位の勘定科目から下位の項目まで体系的・階層的に配置された表をコンピュータの記憶空間に写像する場合は、階層構造を扱うことのできるデータベース・マネジメント・システム（DBMS）、PLI、COBOL等の汎用プログラミング言語のデータ記述機能を用いて財務諸表データの構造化を行うことは容易である。

しかし、貸借対照表と損益計算書や利益処分計算書の項目との横の関係で情報を捉える場合は、何らかの方法によって相互の表を関係づけておく必要がある。

アドレス・ポインタの機能を持つ言語システムならば表相互をポインタで関係づけることが可能で、汎用プログラミング言語を用いても言語自身のデータ定義機能を拡張したデータ記述を行うことができる。

財務諸表の実用システムとしてのデータ構造をポインタ機能を用いて記述するとすれば、リストの発展形である“多分木（multiway-tree）”の概念を用いて階層的に構成されている勘定項目を記述し、表相互の関連づけは多分木のみでは表現出来ないためそのノード相互の横のリンクによる多分木機能の拡張、項目（ノード）索引情報の付加等によってデータ構造の記述を行う必要があり複雑な処理構造となる。

ここでは、下図に示す貸借対照表の複数財務項目の線形リストに関してその基本概念の把握と記述上の課題について考察する。



このリスト構造を定義し、リストの枠の中に財務データを入力する手続きをPLI言語で書くと次のようになる。データは1項目に三つの情報（項目コード・項目名・データ）を含め、これを1レコードとして入力しリストの一つの枠

に收容する。

レコードの終わりの検出によってその直前のレコードをリストの終端枠とみなし枠の前方ポインター部分に終端マーク(NULL)をいれることにより線形リストが完成する。

#### 《記》 PL/I プログラムの表記について

- 文(機能部)は大文字, 他は小文字で表示する (PL/I 言語には大・小文字の区分はない)。
- 記号 “/\*” と “\*/” に囲まれる文字列は注釈でその最初に書かれる数字は文の説明のための文番号とする。構造体宣言 (DCL) の各行の左に書かれている数字は, 構造体内の項目の上位・下位を示すレベル番号である。
- “…………” 部分は, 他の命令文が続くことを表す。
- ここで記述するプログラムは分かり易さを重点にし処理効率は考慮されていない。

#### (1) リスト構造の記述

```

DCL 1 bs    based(p), /* 1 bs はポインター(p)付きの変数で宣言 */
    2 fp    pointer, /* 2 fp は前方の枠にリンクするポインター */
    2 accdata,      /* 3 accdata : 財務項目のグループ名 */
    3 code  char(14), /* 4 code : 財務項目コード */
    3 item  char(80), /* 5 item : 財務項目名 */
    3 data  decimal, /* 6 data : 財務項目データ */
    (q,first) pointer; /* 7
/* q : 後方の枠を指示するポインター */
ALLOCATE bs; /* 8 構造体(bs)の割当 */

```

```

first,q = p; /* 9 第1世代のポインタを first,q に保存 */
ON endfile(sysin) goto lbl2; /* 10 データの終わりで制御を変更 */
lbl1: ALLOCATE bs set(q->fp); /* 11 構造体(bs)の割当(次世代) */
GET edit(accddata) (a(14),a(80),f(8));
/* 12 構造体の財務項目データ部分にデータを入力 */
q = q->fp; /* 13 新世代の構造体領域を示すポインタを更新 */
.....
goto lbl1; /* 14 繰り返し処理のため lbl1 (ラベル) に飛ぶ */
lbl2: q->fp = NULL; /* 15 終端枠のポインタ部にNULLをセット */
k=allocation(bs); /* 16 allocation: 世代の深さを返す関数 */
DO while(k>0); free bs; k=allocation(bs); end;
/* 17 押し下げられた世代を全て解放する */

```

文1のデータ構造名 `bs` は基底(based)付きの変数名で、記憶域が割り当てられたとき括弧内の `p` (暗黙的にポインタ変数として宣言される) は `bs` の位置を示す。このポインタ変数 `p` は、自分自身(`bs`)の位置を外に与えるのみでなく独立した他の構造体の位置(ポイント)を受け入れることが出来る。ひとつの応用例を示すにすぎないが、個々に独立した複数の構造体同志をオーバーラップさせること、記憶域を指し示すポインタの値を変えることによって仮想的なデータ転送を行わせる等、ポインタの性質をいろいろな面に利用していくことが出来る。

たとえば、今期と複数の過去の貸借対照表を別個の記憶域に配置し財務状況の比較分析をする場合、個々の表の位置を示すポインタの値を持つ表を参照し、読み取りをする貸借対照表の該当ポインタを設定することによって、一つの構造体宣言(データ構造)で定義された名前それぞれのデータに接近することが出来る。

同様に次に示すように別個の構造体を宣言しそれをポインターで重ね合わせることによって、例えば、項目コードが組別コードで構成（会社コード・業種コード・項目コード）されていれば一つの記録を別の名前で使うことが出来る。

財務項目の欄についても会社名、勘定項目に区切って別の名前でアクセスすることが出来る。そのための手続きは、単に“p=p2;”とすることにより実現される。

```
DCL 1 bs1 based(p2),
    2 fp2 pointer,
    2 accdata,
    3 code, 4 codec char(4), 4 codeg char(4), 4 codei char(6),
    3 item, 4 coname char(30), 4 itemname char(50);
```

文2の変数名 fp はポインター変数として宣言され、リストの次の枠にリンクする前方ポインターとして使われる。

文3～文6はファクト・データを入れる部分で階層として宣言されている。個々の項目の定義は、P L I 言語の場合では階層のレベル、項目名、データの型とそのサイズの順で書かれる。この例では欄を予め定めた長さで指定しているが、項目をコントロール属性（制御変数）として宣言しておけば記憶域割り当て文によって欄を自由な長さで定義することも出来る。

文7の q はこの構造体とは独立したポインター変数で宣言されている。このポインター変数は、処理している構造体のカレント（現）世代を表すために使われる。

文8は、構造体 bs に記憶域を割り当てる文である。リストの出発点の枠になる記憶域が割り当てられる。

文9は、現世代（第1世代）を表しているポインター変数  $p$  の値を、リストの開始点を保存しておくポインター（first）と、一時的記憶として使われるポインター変数（ $q$ ）に入れておく。

文10は、PLI言語では“ON文”と呼ばれ各種のハードウェアあるいはソフトウェアの条件をチェックし割り込みをかける働きをする。この例の文では、ON文以後に書かれている `sysin` というファイルから入力するレコードがなくなると条件が真となり ON文の中で書かれている文に制御を渡す。

文11は、構造体 `bs` に記憶域を割り当てる文である。set オプションを付けることによってその括弧内に書かれるポインター変数に、構造体（`bs`）に割り当てられた記憶域の位置（アドレス）がセットされる。

括弧内の式  $(q \rightarrow fp)$  はポインターの示す位置を表す独特の表現で複合記号“->”を用いて書かれる。プログラミング言語Cや、ある種のオブジェクト指向言語でもこの書き方がされている。

この式で書かれている意味は、ポインター変数  $q$  の値でポイントされる `fp` の世代を表している。

この括弧内の式で何が行われたかといえば、 $q$  の指し示す世代のポインター変数 `fp` に当該文（`ALLOCATE`）で割り当てられた構造体（`bs`）のアドレスの値がセットされたということである。上のリストの構造図で説明すると、出発点の枠の前方ポインターを入れる部分に当該文で割り当てた2番目の枠の位置を入れること、すなわち枠同志が前方方向に接続されたことを意味する。

文12は、外部ファイル（`sysin`）に入れられているファクト・データを読み取り、グループ名 `accdata` に読取フォーマットを指定して入力する文。

文13は、文11で説明したように現世代の構造体のアドレスを一時的にポインター変数に入れておく文。

文14によりファイルのデータが尽きるまで繰り返され、次々に新しい枠がつけられ枠がつながれていく。

文15は、リストの終端処理を行っている。NULL は関数として登録されているもので 16 進数で “00” を返す関数で、リスト処理ではリストの終端を表す情報としてよく用いられている。

上のリストの宣言と、それに収録されたレコードを呼び出し印刷する手続きを次に示す。

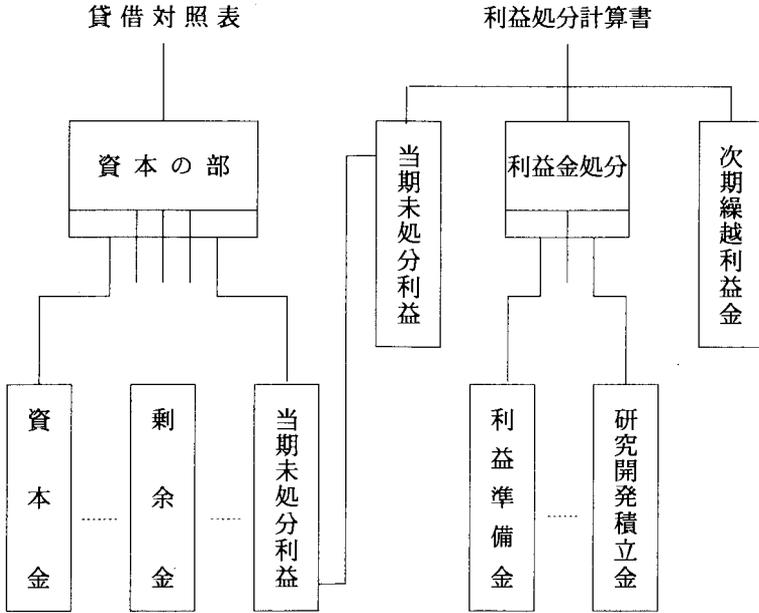
(2) リストに収録されたデータの呼び出し

```
p = first;          /* 1 リストの開始点を p にセット */
lbl3: DO while(p≠NULL); /* 2 処理をNULL検出迄繰り返し制御 */
put list(accddata) skip; /* 3 データの印刷出力 */
p = fp; end;       /* 4 前方ポインターを p にセット */
```

財務諸表は商法や企業会計原則に準拠して作成・報告する義務をもつが、勘定科目の区分・配列方法・分類方法等に関して業種、あるいは企業個々に特殊性を持つ。従ってそのデータ構造は、財務諸表の規範に準拠した表作成の制約と利用面から要請されるニーズの受け入れという両側面の課題に対して柔軟に対応出来ることが求められる。

ここで取り上げた、リスト構造記述のベースになっている“ポインター”の考え方は、複雑なデータ構造表現の理論的基盤になっており人工知能・知識情報処理システムの記述にも用いられる。

次の図は、貸借対照表の一部と利益処分計算書のデータ構造の関係を多分木リストをベースに関係項目のリンクを付加（当期末処分利益金）したものでありポインターを用いて記述することが出来る。



#### 2. 4 抽象データ型と財務諸表データ

データ構造は、前章のBNF表記法を用いて示したように汎用プログラミング言語のデータ宣言分（定義分）を用いて実現することが出来る。

データ構造の定義は、文字（文字列）・数値（整数，実数等）・論理データ等の基本データ型を構成要素としプログラミング言語に用意された記号，キーワード，命令文を用いて行われる。構造化されたデータのうち最も単純なデータ型は、基本データ型の集合で表される“配列”である。また、異なる型のデータ，配列等を階層的に構成して造られる“構造体”等は近代的プログラミング言語ならば、その言語自身のデータ定義機能のみで記述することが出来る。

さらに、ポインター型（指標型）<sup>(2.4)</sup>機能を持つプログラミング言語ならば、ポインターを操作するアルゴリズムとの合体によって、データ構造の代表的な型である“リスト構造”をはじめその応用である“ツリー（木）構造”，“待ち行列型”，“スタック型”等の複雑なデータ構造を記述することが出来る。

しかし、ツリー構造、関係型等のデータ構造を万能型の汎用プログラミング言語を用いて実現するには厳密なアルゴリズムの記述と、単体リスト構造等ミクロな機能から木構造等マクロな機能へと積み上げていくことが必要である。これらのライブラリーの蓄積がないとシステム構築にマンパワーと期間を必要とすることになる。機能を、ツリー（階層）構造型、網型とか関係型等に特定しその型のデータ記述を容易にし、且つデータのアクセスと更新機能も含めデータ操作、作表等を統合したシステムが電子計算機メーカーやソフトウェアハウスの提供する“データベース管理システム（DBMS）”である。

近来、データ構造に関する研究のみならずプログラミング言語、オブジェクト指向プログラミング、AIの研究で注目されているのが“データ抽象化”の問題である。本章次項以降はこれらの視点を念頭におき、問題領域をデータ構造の側面から“抽象データ型（abstract data type）”に限定し考察を進めることにする。

#### 2. 4. 1 抽象データ型について

抽象データ型は、データ定義を行う部分とデータ操作部分を合体させてカプ

---

(2.4) プログラミング言語で用いる変数やファイルの参照をそれが記憶されるアドレスで行えるようにした方式で、そのアドレスを制御する変数をポインター変数と呼んでいる。例えば、C言語では変数名“x”のアドレスは“&x”で表されるのでポインター変数として“px”が定義してあれば次式で x のアドレスが設定可能であり、ポインター変数を用いて記憶域の複雑な制御を行うことが出来る。

```
px = &x;
```

セル化し外部（利用側）に対して内部情報を隠蔽（information hiding）するデータ型で、アクセスや定義はインタフェースのみによって行われる。

オブジェクト指向プログラミングで言われる“オブジェクト”とは、抽象データ型を用いて宣言されたモジュールを言い、また、オブジェクトがあるクラスに属する場合そのオブジェクトを“インスタンス”と呼んでいる。

抽象データ型の登場はデータ記述（データ構造）の理論、とりわけプログラミング言語に大きな変革をもたらした。抽象データ型を最初に取り入れた言語は、1968年にノルウェー計算センター Nygaard,K. と Dahl,O.J. の開発した“Simula”といわれ、以後“Concurrent Pascal”, “Modula-2”, “Ada”, “Smalltalk”, “Objective-C”と受け継がれ発展して来ている。

最近では、「オブジェクト指向プログラミング」のプログラミングパラダイムを与えるものとして汎用プログラミング言語“C++”が注目を集めている。

C++ は、Simula のクラス概念を受け継ぎその機能を拡張発展させた汎用プログラミング言語で1983年にA T & Tベル研究所の Stroustrup,B. が開発した記述言語である。

C++で定義される抽象データ型は“クラス”と呼ばれ、命令文 class で定義され抽象データ型を含めデータ構造の様々な型の記述と、データ構造化、データのアクセス管理等のためのプロセス（C言語ではメンバ関数と呼んでいる）を class 命令文の中で記述することが出来る。class は次のような機能を備えている。

- (a) C言語が有する構文のデータ構造に関する部分の全ての記述機能を持ち、それにC++独自の記述機能が追加される。
- (b) メンバ関数を class の中に持つことが出来その中でデータ構造を定義するために必要な手続きや操作、あるいは、分析のアルゴリズムを記述することが出来る。

抽象データ型が登場する以前はデータ構造の定義とプロセスはそれぞれ

れ独立しており、データ処理を記述するにはそれらの関係に注視し相互の整合性を取る必要があった。

- (c) データのアクセス管理（データ保護、データの隠蔽等）をクラス自身の中でアクセス指定子（private, protected, public）を用いて行うことが出来る。また、指定した特定の相手のみアクセスを許す手続きを記述する friend 指定を class の中で指定しておけば、指定された者のみフレンド関数を用いてその class で定義されたデータに接近することが出来る。
- (d) データ構造を定義した一つのクラス（基底クラスと呼ぶ）から必要な部分を受け継ぎ、その機能に加えて拡張したクラス（派生クラス）を導出する機能を持つ。オブジェクト・プログラミングではこのことを“継承”(inheritance) と呼んでいる。複数の基底クラスから一つの派生クラスをつくる場合を多重継承と呼び新しい処理系ではこの機能を備えている。

継承機能の利用によってデータ構造としては複雑な形を持つ多階層木構造も比較的容易に実現することが出来る。

- (e) 記号（演算子）や関数を既に定義付けられている仕様とは異なる仕様で再定義する機能を持つ。これを多重定義（overload）と呼び関数の場合を例にとれば、その仮引数（パラメータ）の型や数を変えて定義しておけば一つの関数を柔軟に引用することが出来る。関数を呼び出したときにそれを特定するのは処理系の役割で仮引数の数と型をチェック選択し一致した定義の仕様で呼び出す。

抽象データ型は従来のデータ表現を根底から変えるものである。プログラミング言語の中で定義されたクラスは一つのデータ型と見なされるため、理論的には無限のデータ型の表現を汎用プログラミング言語で行うことが出来ることを意味する。

所謂「オブジェクト指向プログラミング」と言われているのは、抽象データ型のデータ構造によるデータの抽象化と通常の手続きの抽象化を一層進め、そこから生成されたオブジェクトの集合でプログラムを組み立てることである。

#### 2. 4. 2 抽象データ型と財務諸表<sup>(2.5)</sup>

発生した取引を記録する複式簿記体系を通して作成される財務諸表は、企業の財務体質をあらわす情報として企業の意思決定に役立てられ、また、株主、投資家、銀行等外部へ企業の財務状況を伝える情報として重要な資料である。

財務諸表は、国・業種・企業等によって特殊性を持っているものの商法・企業会計原則等でその計算方法・仕訳・分類・報告様式等が定められているためデータ構造を記述する対象としては適切な分野でもある。

財務諸表と言われているのは次のものである。

- (1) 損益計算書(PL:profit and loss statement)
- (2) 貸借対照表(BS:balance sheet)
- (3) 利益金処分計算書(SA:surplus appropriation statement)

損益計算書は、ある営業期間内に発生した収益とそれに対応した費用とを一つの表にしたもので、そのデータは企業の収益性を表す。

貸借対照表は期末の資産、負債、資本を一覧表にしたもので、その表は複式簿記では表の左側に“借方”，右側に“貸方”を配置した表である。借方は資産で、貸方には負債・資本（当期末処分利益を含む）の各会計項目である。資

---

(2.5) ここで示すデータ構造のモデルは、財務諸表全体から考察したものではない。対象とする表、勘定項目も限定されたものでありデータ構造を記述することに着目して組み立てられたFS(feasibility study)的試論である。雛形モデル、実用モデルへの展開は、財務理論、財務諸表に関する専門家を交えた検討が必要である。

産はまた流動資産・固定資産・繰延資産に分けられ、各々の項目は更に細分化される。負債・資本の部についても同様である。

利益金処分計算書は、当期末処分利益金の処分内容、および次期への繰越利益金を表す表である。

財務諸表のデータ構造の設計に於ける要件は、報告義務の有る表、各種計算書の作成は当然のことであるが、重要な点は各種の財務諸表分析（財務構造、損益構造、原価構造、付加価値構造等の分析）一つひとつの分析が効率的に行え且つ利用面からの多様なニーズに対しバランスのとれた総合分析を行い得るデータ構造を考えていくことである。

従って、ここでは財務諸表分析を財務理論から捉えるのではなくデータ構造に着目しながら計算構造の面で区分し、その分析を行うためには如何なるデータのアクセス法、データ操作、データ処理のアルゴリズムがデータ構造の記述面に要請せられるかを考察する。

#### (1) 各種財務諸表を記述する情報（メタデータ）

データそのものを記述するデータでデータをレベルで分けるとすれば一階層上のレベルのデータをメタデータと呼んでいる。データベースの“データ辞書”はこれと同義的に使われる用語である。

財務諸表データを対象としたメタデータは、データ構造全体を制御するデータ定義情報を含みその主なものを次に示す。

- (a) 各種のデータ操作、分析のアルゴリズムが蓄えられている場所の所在を表わしたディレクトリ。
- (b) 貸借対照表・各種計算書等、表を管理単位とする表相互の関係を記述した情報（表内・表外項目とのリンク情報等）。
- (c) 表・計算書自身が持つ情報（勘定項目名、借方・貸方の区分、フローデータ・ストックデータ等）。

(d) データのアクセス権限を与えるセキュリティー情報。

#### 《データ構造に関する注釈》

ここで扱われるデータは全て表が管理単位となるため表構造の定義機能、表のアクセス手段が用意される必要がある。

さらに、メタデータは追加・更新される情報であるため階層関係を含めメタデータそのものの定義・記述が必要である。

### (2) 財務諸表の入力、各種計算書・報告書等の作成情報

損益計算書、利益金処分計算書のデータは期間中のフローデータで、貸借対照表は期末のストックデータを表し、次に示すような表構造上の特徴を持つ。

#### ・項目の階層関係

勘定項目は次の例のように階層関係で表される。

$$\text{売上高} = \text{製品売上高} + \text{商品売上高} + \text{その他の収益}$$

#### ・表、計算書相互の階層関係

例えば、損益計算書に上げられている売上原価の中の当期製品製造原価は“製造原価明細書”によって算出された勘定項目である。

#### ・表、計算書の項目参照

表・計算書で得られた勘定項目が他の複数の表・計算書に一つの項目として上げられる場合がある。

たとえば、損益計算書で求められた“当期末処分利益”は利益処分計算書および、貸借対照表の資本の部の1項目として用いられる。

#### 《データ構造に関する注釈》

表内の勘定項目、表相互の関係は階層で表されるのでデータ構造も階層構造が基盤になる。表相互の結合は、ポインターを用いたリスト構造を用いれば簡単に表すことが出来る。階層構造の他の表（抽象データ型

での他のクラス)の勘定項目を書き換える場合はアクセス権限に従う何らかのルールが必要である。

上の例でいえば、損益計算書で求められた“当期未処分利益”を貸借対照表の一項目として書き込む場合である。この操作は、抽象データ型の機能の一つであるフレンド関数を用いることにより容易に実現することができる。

### (3) 比率分析に関連する情報

比率分析は伝統的な経営分析の手法で特定時点の決算期のデータを基に企業の財務流動性・収益性・安全性・生産性等が分析される。

#### 《データ構造に関する注釈》

表・計算書自身に収録された勘定項目を用いて所定の計算式により比率を算出し、結果の出力仕様もここで記述される。

また、比率の結果により何らかのアクションをとる必要があり、そのルーチン化が可能であれば、その手続きのアルゴリズムを組み込むことにより分析とデータの整合をとることが出来る。

これらのデータ処理は、抽象データ型の中で定義するメンバー関数で行われる。

### (4) 比較財務諸表分析、及び、趨勢財務諸表分析に関連する情報

比較財務諸表分析は、過去数期の決算期から今期までのデータを抽出・比較することによって財務データの動きを測定する分析である。趨勢財務諸表分析は特定の決算期のデータを基準にし他の年度のデータを基準年度の比率で表し財務構造の推移を相対的なデータでもって比較・分析しようとするものである。

#### 《データ構造に関する注釈》

財務諸表データの性格から、それらの表・計算書がデータ構造としてコンピュータの記憶域に実装される単位は、特定の決算期で特定の表ないし計算書ということになる。

したがって、比較財務諸表分析・趨勢財務諸表分析を行うには異なった決算期のデータがアクセス出来る表構造、たとえば、各々の表に時間軸を入れて実装単位を3次元の表にするか、2次元の表相互に表のリンク情報を付加しそれを手がかりにしてデータをアクセス出来る仕組みにしておく必要がある。コンピュータのリソースの効率的利用とソフトな構造（柔構造）のデータ構造を構成できることから後者が選択されよう。

データ構造は、複数の表・計算書（抽象データ型では基底クラス）からデータを受け継ぎ（多重継承）、分析手法は複数の分析手法をメンバー関数の中で記述・登録することが出来る。分析手法の選択と情報の伝達は、そのクラスの中のメンバー関数が備え持つパラメータに指定情報を与えて所要の“オブジェクト”を生成し、そのオブジェクトを呼ぶことで処理が遂行される（オブジェクトは抽象データ型を用いて宣言されたモジュール）。

#### (5) 統計解析による総合的分析に関連する情報

総合的分析では次の二つの側面で分析が行われる。

##### a) 横断面分析 (cross-section analysis)

特定の時点の財務データに関して他企業、同種産業平均等同一集団に属する複数組の原データ・財務比率データを入力し統計的手法を用いて行われる分析である。この分析では多変量解析の手法が一般に用いられる。判別分析による“アルトマンの企業破産予測モデル”，多数の説明要因をいくつかの成分（収益性，安全性，成長性等）に要約して分析する主成分分析等が用いられている。

## b) 時系列分析 (time-series analysis)

財務データを時間系列として捉える分析で、説明変数を移動平均、季節調整で原データに修正を施すこと、説明要因を時間の関数で表したモデルをつくって最小自乗法等を用いて分析される。この分析での入力データは利子率で割引かれたデータ（キャッシュフロー）が用いられる。

## 《データ構造に関する注釈》

上の総合的分析で用いられるデータの対象は自社のデータに加えて、他企業及び産業平均の財務諸表データ、更に金利・物価指数等の一般データが分析に用いられるということ、また、分析手法も特定出来ないと思われるのでデータと分析アルゴリズムをバインドしたデータ構造として定義することには多くの困難を伴う。

データ構造の一つのモデルとして考えられるのは、データ構造全体を記述制御するメタ情報（表・計算書の共通情報）の中から指定する情報と指定する分析手法のディレクトリ情報を、抽象データ型の特徴である“継承機能”で当該クラスに引継ぎ、分析で用いられる特殊データに関しては当クラスのデータ定義部で行い、また、継承で受けたアルゴリズムの情報を含めて所要の分析を行うための論理を、当クラスのメンバ関数で記述する方法である。

### 2. 4. 3 新しいデータ型の課題

オブジェクト指向プログラミングは、情報化の進展によって増え続けるコンピュータの処理に対する要求とそれに応えるプログラムの生産性の不均衡を改善するために、新しいプログラム・パラダイムの下でヒューマンインタフェース等の新世代のニーズに対応したシステムを構築する手法でもあると言える。

本章では、オブジェクト指向プログラミングの考え方の基盤ともなっている

データ構造（抽象データ型）に着目し、その基本概念の把握と財務諸表データへの適用可能性に関して、システム分析面から若干の検討を行い次に示す成果と課題の提示を行うことが出来た。

財務諸表のデータ構造をデータ記述の論理的側面に視点をあて考察した（十分なサーベイとはいえないが）結果、その構造は単純ではなく“表”，“階層”，“網”等の組み合わせより構成される複合構造が要請されるということ、そのような背景に対して、データ（表）と財務分析の一体化を計った抽象データ型の財務諸表データ構造への適用に期待が持てることが判った。それを発展させることによって、報告としての財務諸表と財務分析とを有機的に関係づけた財務システム構築への道も開けて来よう。

今後の課題としては、実際のモデルを構築・検証すると共にデータの対象を広げ総合システムへと展開させることである。

## 2. 5 新データ構造を組み入れた情報分析システム

本章では、財務諸表データを例にしそのデータ構造のモデルとして“リスト構造”，および“抽象データ型”を取り上げ実用システムへの適用可能性について検討してきた。

リストをデータ構造に取り入れたデータ処理は、お互いに関連を持つデータ項目（レコード、フィールド等）の構造化・データ操作に効果が期待できる反面プログラミングの側面からみると複雑な処理を伴う。

それらの手続きを簡単にまとめると次のような手順を必要とする。

- 構造（線形、環状、二分木、多分木等）を定める。
- リストの枠に関する定義  
箱、前・後方ポインター、一方・双方向リスト、データへのリンク等。
- 適用対象のデータに対するリスト構造の組立

根 (root)・節 (node)・枝 (branch or pointer)・葉 (leaf) 等の箱に関する記述, 親 (parent) と子 (child) の関係記述等。

・リスト処理 (操作)

リストの挿入, 削除, 反転 (ポインタの付け替え), 探索等。

リスト処理のプログラミングのツールは, 汎用プログラミング言語, または, リスト処理を効率的に行う LISP<sup>(2.6)</sup> 言語 (族) の何れかによって行われる。

前者の言語を用いて記述する例 (財務データ) を本章 (2.3) に示したのがかなり煩雑な記述を必要とする。後者の言語を用いれば高度の記号処理を行うことが出来るがエンド・ユーザにとっては大きな負担となる。

複雑なリスト処理をエンド・ユーザが容易に操作することも可能であるので, 一つの方法としてその枠組みについてのみ簡単に触れることにする。

(a) リストの箱・節・枝等, 形で表現できる物 (パーツ, リンケージパス等) を図形 (グラフ) で表す。これを“ $\alpha$ 図形”と呼ぶことにする。

階層構造・網構造とその組み合わせで表される組織図・各種系統図・フローチャート等はグラフで表示することによりその形を直感的に認識することが出来るばかりでなく, グラフの編集 (箱の作成・挿入・削除, リンケージパスの操作等) もその全体とグラフ相互の関係が明確に確認出来るためマウス等によって図形を操作することが出来れば容易に行うことが出来る。

(b) エンド・ユーザの操作によって $\alpha$ 図形が更新された場合, システムはシステム機能によって“ $\beta$ 言語”を用いてリストで表されるデータ構造を

---

(2.6) LISP: 1959年 MIT の John MaCarthy によって考案された記号処理言語。数多くの同族の処理言語が開発されているが, 現在言語の標準化を進めた“Common Lisp”に統一されてきている。

更新する。 $\beta$ 言語は情報分析システムを記述した言語、または LISP 等の記号処理言語である。何れの言語を用いる場合でも各種のリスト処理を行うサブプログラム・ライブラリーの整備はシステム開発担当者がしておく必要がある。

(c) エンド・ユーザが行う分析操作は、“ $\gamma$ 言語”が用いられる。

$\gamma$ 言語は、情報分析システムの目的言語である。SECRETARY ならば、リスト処理を行うために追加されたコマンド集合である。

次に、抽象データ型を用いて宣言されるオブジェクト（モジュール）機能を情報分析システムに取り入れる問題であるが、その機能を目的言語のコマンドとして組み込むことは次の方法を取ることによって可能である。

それは情報分析システムの記述言語に抽象データ型を支援するプログラミング言語を選択することで実現できる。現時点では、C++、Concurrent Pascal<sup>(2.7)</sup>等がその条件を満たしている。オブジェクト指向プログラミングを可能とするプログラム開発基盤の提供は、高度の知識と関連する膨大なプログラム集合の蓄積がなくては出来ないことにより汎用プログラミング言語が選択される。

---

(2.7) Concurrent Pascal : 1974年にブリック・ハンセン (Brink Hansen, P.) が考案したプログラミング言語。



## 第3章 財務データベース構築の基盤

—収録データと管理・運用の基盤について—

財務データベースに関しては、章を分けて二つの視点から考察する。

本章では、財務データベース構築に要請される基本的な問題と管理・運用的側面に関して触れ、併せて財務情報分析で必要とされるデータ対象を探る。

財務データベースの構造的問題に関しては、ネットワークとの関連で第4章で扱う。

### 3.1 データベースの発展と構築の要件

データベースの概念と発展史は次のような視点で捉えられている。

- (a) データベースの原形は、米・英にみられる資料保管所 (Data archives) であるといわれている。

データアーカイブは、主として学術的ないし研究・調査の要求から出発したもので、第一段階では“analyst researcher”の要求にそうため特定のデータを集めたことに始まったとされている。<sup>(3.1)</sup>

データアーカイブに関しては、「統計データセンターの基本的考察」<sup>(3.2)</sup>の中で述べられている。

- (b) その後コンピュータを用い、ファイルを統合化し、データの管理機能を持たせたシステムが1950年代の終わりから1960年代の始めにかけて米軍によって“管理指令システム”の名で開発された。そのシステムをデータベースと呼んだのが初めてだとされている。<sup>(3.3)</sup>

その後、データベースの定義・制御・操作を行う“汎用データベース・マネジメント・システム (DBMS)”の好ましい方向付けへの要請のもとに

“CODASYL (Conference on Data Systems Languages) システム委員会” の調査活動が行われその調査報告書が刊行されている。

この委員会での調査目的は、下記の通り記されている。(3.4)

- 汎用データベース・マネジメント・システムを記述するための共通の基盤となるものを見いだすこと。
- 有用さを啓発すること。
- コンピュータ業界に、この情報を提起すること。

CODASYLの活動は、データベースの基本概念・規範を世に示し啓蒙活動によってその発展を促しデータベースの方向付けと多くの実績を残した。

(c) 1970年コッド (Codd, Edgar F.) の提唱した“関係モデル”を契機に、ALPHA, SEQUEL, QBE, SQL等の関係データベース言語、および、System R (IBM), Ingres (カルフォルニア大学バークレー校)等の“関係データベース・システム (relational database system)”，が開発されデータベースは急速に発展していった。現在のネットワーク環境の下に構築されているデータベース・システムの殆どは関係データベースを基盤とするものである。

現在データベース (もしくはデータバンク) は、情報管理・情報サービスの核としてネットワークと共に情報システムの基盤を形成している。

データベースのサービスを主目的としたシステムでは、官公庁が公共事業の推進・サービス提供のためにすすめるデータベース、営利企業の商用データベ

(3.1) 藤川正信, 「経済情報の理論と管理の方法論」, 行政とADP, 1971.5, p.24.

(3.2) 生島芳郎, 「統計データセンターの基本的考察」, 経済経営研究叢書, 経営機械化シリーズ 13, 神戸大学経済経営研究所, 1972, pp.103-110.

(3.3) 水野幸男・内田幸久, 「汎用データベース・システム」, Computer Report, 1971.5, p.44.

(3.4) 小林功武編著, 「データ・ベースの管理」, 企画センター, 1970.7.

ース、学術研究のためのデータベース等々、収録分野、利用対象、利用目的に沿った種々のデータベースが構築されている。

一般に、データベースを構築し利用に供していくには次に上げるような制度・組織・技術的な基盤整備、および、そのシステム開発・運用に関わる費用の保障が必要となる。

#### (1) サービス提供のための設備等の資源整備

データ収集・整備・蓄積・保管，サービス（オンライン端末，ハードコピー装置，磁気テープ等の記憶媒体等）を提供する設備の整備。

#### (2) 管理・運営組織と監査

提供データの収集・選択，公開・非公開データの区分け，スクリーニング・サービス，データのセキュリティ管理，等受益者サービスに関する基本的な問題の審議と決定，データとサービスに対する利用者側要求の集約・対応組織，提供データの著作権の問題，予算案作成，決算報告・システムの監査，システムの評価・見直しを行う等データベースの管理・運営を行っていく組織。

#### (3) 管理・運営，およびデータ提供のためのソフトウェア整備

データベースを支援するソフトウェアとしては，データの蓄積とメンテナンスを行うデータベース管理システムの整備。

利用者に対するサービス・システムとしては，利用目的に合わせたデータの多様な検索手法の提供，データの操作・分類・加工（演算）・分析手法（各種分析ルーチン），結果・報告書出力機能，データおよびサブプログラムを含む実行用プログラムのカタログ化とそのメンテナンス機能等の提供。

#### (4) 利用者に対するサービス

データベース利用のための設備，ソフトウェアのインストール等に関

する支援，利用マニュアル，ニュース，利用相談等利用に関わるサービスの提供。

### 3.2 財務データベースの特色と収録データ

前項で触れたようにデータベースを構築し利用に供するには，組織・運営上の問題とそれを支援していく技術的問題（主としてソフトウェア）が存在する。本叢書で対象とするデータベースは，企業の財務データを主要なデータ（公表される貸借対照表・損益決算書等の財務諸表）とし，その他官公庁で出される企業に関する統計データも収録の対象に含める。

データベース利用者は，これらのデータを素材にし企業内における投資・予算編成，決算・連結決算等のためのデータ分析・資料作成等部内ユーザの利用から，企業集団（取引・資本系列）・業種等でアグリゲートしたデータを用いてその財務状況・成長性等企業経営のグローバルな視点からの財務分析を含め企業の意思決定者が求める情報の取得に至るまでの広域にわたる情報サービスをデータベースから受けることを想定し，そこで必要とされる財務データの対象を探ってみることにする。

#### 3.2.1 財務諸表データ

財務データのサービスは，現在次の三つの媒体で提供されている。

- (a) 磁気テープ，フロッピー・ディスク，CD-ROM等の高密度媒体
- (b) ハードコピー・サービス
- (c) オンライン・サービス（TSS等）

ハードコピーとオンライン・サービスに関しては，データ加工・簡単な分析等利用者の要求に合わせたサービスも含まれる。

オンライン・サービスの場合は、サーバにオンラインで接続しサーバの提供するデータ検索・分析用対話型コマンドを用いて利用する。

(b) , (c) のサービス・メニュー等に関しては本項ではこれ以上触れない。

貸借対照表・損益決算書等の財務諸表の磁気テープによる一般ユーザへの提供サービスはかなり古くから行われてきた。

日本では、日本興業銀行系列会社の(株)興銀データサービス、日本経済新聞社(データバンク局)等で提供サービスが行われている。

外国では、米国S & P社(Standard & Pool's Compustat Service, Inc.)の“COMPUSTAT”は米国における財務関連データに関する代表的なサービス機関としてよく知られており、英国E x t e l社の“EXSTAT”は自国を含め他の20カ国の財務データを収録し提供サービスを行っている。

これらの提供データファイルのうち、本研究所“国際経済経営データベース(BEICA)”に収録のもののデータ仕様の概要を以下に上げる。

#### 《興銀財務データ》

興銀データサービス提供の財務データは収録項目数(本決算)が多く(約400項目)、貸借対照表・損益計算書・利益処分計算書・製造原価明細書のいわゆる財務諸表の外、明細項目として株式の状況・従業員の状況・監査意見・棚卸資産の基準及び方法・有価証券明細・減価償却明細・社債の変動・長期借入金の変動・外貨建資産負債明細・担保差入資産・現預金の内訳等の項目。

また、年換算前のファイルについては売上の主要品目について品目別の売上金額も含まれている。

さらに、電力・ガス・水道・運輸・通信・海運業に関しては業種特有の項目も収録されている。

現在磁気テープで提供されている興銀財務データは、商法改正への対応と将来の会計制度の大幅な変更にも即応し得るものとして1987年9月に全面改訂されたフォーマットのものである。

それ以前の旧フォーマットのファイルでは、1964年より30期収録されている半期・年決算混在ファイル、1966年より20期収録年換算ファイルがある。

〈資料〉「項目コード表」、興銀データサービス、

### 《日経財務データ》

#### (1) 日経金融財務データ

金融・証券業に関しては貸出金明細・信託勘定等財務項目が異なり一般製造業・サービス業とは別のファイルでまとめられている。

日経金融財務データには、全国銀行として長期信用銀行・都市銀行・地方銀行・信託銀行、および相互銀行の日本の全銀行の財務諸表(250項目余)が収録されている。

項目の中には日経独自に収集した未公表の項目、また、一部の項目については日経財務基準によって組み替え銀行相互間の財務比較がしやすいように編集されている。

〈資料〉「日経金融財務データ(連結決算)」、磁気テープ説明書、日本経済新聞社データバンク局

#### (2) 日経財務データ・連結決算

収録項目は約120項目。収録会社は、銀行・証券・保険を除き大蔵省に連結有価証券報告書を提出している全企業。

〈資料〉「日経財務データ(連結決算)」、磁気テープ説明書、日本経済新聞社データバンク局

## 《S &amp; P社（米国）財務データ》

S & P社は、米国およびカナダの主要企業の財務データ、企業統計データ、マーケット情報等のデータに関して広範囲の業種わたって収集・編集し、企業の財務内容・財務比率・信用・投資等産業・企業・証券等の経済情報に関する資料を刊行している米国有数の経済情報出版社である（1860年創業）。

同社は調査・出版事業のほか、財務データを磁気テープ等に収録し“COMPUS T A T”の名称でデータファイルの提供サービスを行うと共に財務データの利用マニュアルを含めて財務領域の総合サービスをタイムシェアリング・システムを通じて行っている。

収録対象業種は、工業企業（製造業）、および、銀行・生命保険・鉄道輸送等の非工業企業（非製造業）で総収録企業数は4,800社に及ぶ。

収録項目数は、主要項目（175項目）と明細項目（175項目）に分けられその内容は、貸借対照表・損益計算書に関する項目、株価・従業員数・設備投資・研究開発費・資金運用等の特別の項目も含まれている。

〈資料〉「COMPUS T A T」, Standard & Poor's Compustat Services, Inc.

## 《Extel社（英国）財務データ》

欧州では、英国 Extel社の“EX S T A T”が欧州の企業を中心に欧州に進出しているオーストラリア、日本企業等を加えた20カ国、約3,900社の財務諸表を収録している。

Extel社（Extel Statistical Services Limited）は、英国をはじめフランス、ドイツ（西ドイツ）などヨーロッパを中心とする有力企業、および一般の民間企業だけでなく有力国営企業、オーストラリア・日本等の現地法人も含めた企業の財務データをファイル化し“Exstat”の名称でサービスを行っている。

収録企業の概数は、英国 3,900 社、フランス 140 社、ドイツ 130 社、オーストラリア 260 社など 20カ国総計 3,900 企業。

項目では、英国企業 330 項目、英国以外 200 項目が収録されている。

〈資料〉「Exstat Accounting Requirements, Terms & Ratios」, Ex-tel Statistical Services Limited

上の国内外の財務データのうち“興銀財務データ”，および“COMPU-STAT財務データ”に関しては、収録財務項目コード表を本叢書〈付録3〉および〈付録4〉に掲載した。

### 3.2.2 財務関連データ

以上、本研究所データベースに収録の財務諸表関連のデータをリストアップしたがこれは財務関連データの全域をカバーするものではない。しかし、データベースといっても分析で用いられる利用頻度の低いデータを含めあらゆるデータを蓄積・保管することは経済的にも好ましくない。これらのデータは外部のデータベースからインターネット等によりアクセスできるようにし、収録データベース機関に関する所在情報とデータのアクセスの方法をデータベース機関が情報として提供できるようなシステムを採ることで対処出来る。

そのほか、財務データベースとして収録すべき財務諸表以外の主要データを下記に上げておく。

- 株式データ（株価、配当率、発行株式数、株式売買出来高等）。  
このデータは、株式投資決定分析、株価収益率、CAPM（資本資産評価モデル）、ポートフォリオ分析、配当政策、資金調達等の分析・意思決定に用いられる。
- 各種金利・利回り（預金・貸出金利、信託・債券・各種金融商品の金利等）、

投資決定、資金運用、資本コスト、企業評価（分析手法：正味現在価値法、内部収益率法、キャッシュフロー、指標の総合化／多変量解析、残差分析）等の分析に用いられる。

- ・各種物価指数、デフレータ等。

### 3.2.3 財務データの特徴

次に、データベースのファイル設計（データ編成方式・データ定義等）、情報分析システムの設計（コマンド）に重要な情報を与える財務データ（特に財務諸表について）の特徴・特殊性、および利用態様を下記に示す。

前者に関しては、

- (a) 短いスパン（中間決算・本決算）で定期的な作成される。
- (b) 国・上場・業種・企業群・規模等の区分による集団で扱われる場合が多く、分析ケースによりデータの取扱いが異なる。
- (c) 倒産・合併等の動静がありデータの整理編集はマニュアル的になる。
- (d) 収録対象（国・上場・非上場・項目等の区分）を広げると膨大なデータ量となる。

等が挙げられる。

また、財務データの利用態様に関しては次のような特徴を持つ。

- (a) 利用者は、作成当事者（企業内）から利害関係者（アナリスト、金融業者、投資家等）まで広範囲に渡る。

したがって、分析プログラムは、財務諸表の作成支援・財務諸表による内部分析から金融機関・投資家が利用する信用分析・投資分析等の外部分析までに渡る。

- (b) データでは外国と日本、業種全体と当該企業等のデータ整合性の問題があり、分析内容では予算と決算（計画と実績）、基準と実績、等分析

過程で比較・対照論的分析・評価がよく行われる。

- (c) 膨大なデータ対象から分析テーマ毎に幾組ものデータが速やかに検索・編集出来ることが求められる。
- (d) データが最終的に分析アルゴリズムに結合される時はアルゴリズムの要求に合わせたデータの編集・変容が行われる。
- (e) データ操作は、横断面、および時系列両方の扱いがされる。

### 3.3 財務データベース利用マニュアル

はじめに、データベースの提供者が利用者に対して用意しなければならない資料、利用マニュアルについて触れ、続いて前項(3.2.3)にあげた諸々の利用態様に対して解答を与えるための情報分析システムに要求される諸機能について考えてみる。

収録した財務データをシステム化し利用に供するためには、データベースを管理・運営していく組織と、サービス・システムを構築しなければならない。とくに必要となるのが利用者に対するサービスとして“データベース利用マニュアル”の作成であり、その資料整備如何はデータベース評価の重要な基準ともなる。

財務諸表(貸借対照表、損益計算書等)を対象にした財務データの場合は、いくつかの会計規則に基づいて公表されるため、先ずそれに従った会計各項目の定義・計算法と算出根拠・評価法・項目配列・項目相互の関連等々に関する注記・注釈、および利用上の注意事項等のマニュアルは提供者側の義務であり準備されるべき資料である。

また、財務データは国・業種・企業等作成主体の個別性から必然的に生じる会計項目の定義・分類基準の相違、またデータ提供者によって独自の編集が行われる等、提供データそのものに関して特別の扱いが行われるので提供者はそ

れを利用者に周知徹底しておかなければならないし、利用者は利用に際しそのデータの仕様・属性を把握しておく必要がある。

さらに、利用者に対する提供データの利用法、財務諸表の作成方法、財務分析のためのノウハウ、またそのようなサービスを総合的に行う“財務情報分析システム”の提供に関しても財務データベースサービス機関が持つこと望ましく、これら諸々の問題に対して利用者への指針となる利用マニュアルは、データベース・ベンダーが準備しなければならない必須の資料である。

利用マニュアルとしては次のような資料が必要となろう。

#### (1) 提供媒体の仕様書

磁気テープ、フロッピー・ディスク、CD-ROM、MO等マシンリーダブルの媒体による提供の場合は、媒体の物理的仕様、収録情報記録フォーマット、収録フィールドのデータ形式等の諸情報の仕様、および、収録データのプレビュー（キー項目による分類表、チェック・リスト等）を行う簡易印刷出力プログラムの完備とその使用法等の仕様書・説明書類の整備。

オンラインによるサービスの場合は、目的の財務データそのものの検索利用に加えてデータ仕様に関する情報も端末機からアクセス出来るような利用システムが要求される。

#### (2) 収録データのコードブック

財務データを対象とするデータベースの場合は、収録会計項目のコード表、業種コード表（財務データを用いる財務分析では業種区分は重要な意味合いを持つ）、上場の証券コード表、個別企業コード表等。

産業は、第一・二・三次それぞれの産業によって、あるいは、軽工業・重工業、公共的企業・民間企業等によって企業の組織、経営活動、営業評価の規範が異なる。

たとえば、企業の経営動向を視るとき、ある時点での景気の好不況は産業全体の動きの外業種によって特有の傾向が現れる。また、個別企業の経営分

析においてもその企業が属する業種全体のアグリゲートした基準値との比較で経営診断の材料にすることも通常行われる。

業種区分は、J I Sの日本産業分類表も定められているが、現在では財務データ提供者独自の業種コード表を作成しているところが多い（興銀財務データ、COMPSTAT財務データ等）。

これらのコードの外に証券コード、日経金融財務データのように業界で定められたコード情報を提供媒体に含めているものもある。

証券コードは、上場企業個々にコードが割り振られているが組別コード構成の方式に従っているため大まかな分類ではあるがこれによって業種を区分することが出来る。

次に、現代の企業は経営の合理化、高機能・高信頼性の製品提供、顧客への多様なサービスの必要性から、国際協調、変動為替制度下の企業運営、製品の市場、労働力の需給関係等の外的諸要因を含め資本関係を基盤に企業の集団化が進んでいる。企業の財務分析を行う場合に於いてもその必要資料として個別企業の財務諸表のほか企業グループの連結財務諸表が必要となるし、企業グループでの子会社の貢献度をみるとか、親会社と各々の子会社との財務関係を計測したりすることは財務分析の枠組みに入れられるべきであり、企業グループに渡るコード表も必要となつてこよう。

### （3）会計項目の定義表

いわゆる財務諸表と呼ばれているのは、貸借対照表、損益計算書、利益処分計算書でそれに財務諸表付属明細表、営業報告書が加わったものをいう。

これらの諸表は、中間決算、連結決算に関するものも含まれる。

項目に関する情報としては、勘定項目名とそのコード、項目に含まれている勘定科目の内訳を明確に示した資料が求められる。

勘定項目コードのコード体系は、財務諸表の分類構造から通常定められている（興銀財務データ、日経財務データ、Exstat 財務データ等）が、なか

にはコードに一連番号を付し貸借対照表、損益計算書等の財務諸表作成のためのコード・セットを別途資料として提供しているデータベース(S&P社 Compustat)もある。

前者は財務諸表項目の上位・下位概念で組別にコードが付され符号と意味が関係付けられているため分析プログラムで扱い易いという特長を持つ。

問題となるのはコード体系、およびコードそのものの定義が明確にされていることと、各種財務諸表を作成するためのマニュアルとそのコード・セットをデータベースを提供するベンダーが提供することが必要である。

ユーザはそれらの資料によって情報分析を厳密且つ容易に行うことが出来ると共にそのノウハウを受けることにより分析の可能性を広げることが出来る。

#### (4) データ検索・作表, 分析ツール

いままで触れてきた問題は、財務データベース成立の一般的条件、収録データの対象(種別、範囲)は何か、集録データの仕様等データベース構築の管理・運用的側面に関することが中心であった。

データベース提供者の果たすべき役割の第二は、ユーザに対して如何なるインターフェース(機能)を提供出来るかが問われる。

高い生産性と高品質の製品を製造する設備を持つことは技術革新・ノウハウの蓄積に貢献はしてもその設備から生産される製品に需要がなければ生産システムは再生産を行っていくことは出来ない。そのシステムがサービスの提供を目的に構築されたシステムならば公共・商用を問わず受け入れられない。

データベース・システムではそれぞれ異なった目的を持つ各階層の利用者に対して如何なるノウハウ、ツールを提供出来るかが重要な点で、それは利用者が最も関心を持つ部面でもありシステム評価の重要なポイントになる。

ここでは、データベースの利用対象(者)を

- a) 一般利用者（非定型的随時利用）
- b) データベース資源を利用した応用プログラム作成者（アプリケーション・プログラマー）

に分け提供者はそれぞれの利用対象者に如何なるツールを用意すべきかを、利用対象者とその要求機能を実現する情報処理言語ないし分析システムについて簡単に取り上げてみる。

a) 一般利用者

分析システムによる例示に応じていく情報入力操作方法、あるいは、簡単なコマンドを用いて収録データの通覧、必要データの検索・作表・ビジネスグラフの描画をさせること、貸借対照表・損益計算書・利益処分計算書等の財務諸表を指定企業・資本系列・連結会社集団・業種グループ等々の分類で簡単な指令で表示させ得ることは通常の機能として用意されていなければならない。

また、分析の自由度を上げるために対話的に利用できる数式処理機能（コマンド）、および、数学関数（平均・最大値・最小値・標準偏差・相関係数等）、財務関数（終価計算・現在価値・内部収益率・ローン計算等）、減価償却計算等が情報分析システムに組み込まれ、これらの関数・ツール類が単独あるいは数式処理コマンドで引用出来る分析用言語の提供も必要になる。

さらに、利用者が用いるデータおよび分析者固有の分析手法の登録・編集・引用が利用者自身で出来ること等が最低限の機能として望まれる。

b) データベース資源を利用した応用プログラム作成者

データベースにアクセスするためのインターフェース仕様書、そのツール群の整備と操作マニュアル。

たとえば、データベースの型が関係型ならば“SQL（データベース検索・操作言語）の提供、ネットワーク環境下でのクライアントの定義・構築のためのユーティリティー等、いわゆるミドル・ソフトウェア(3.5)の提供等

はミニマムの整備条件である。

### 3.4 財務分析の対象データと収録構造

財務諸表は、企業の経営内容を内部およびその利害関係者に伝え、それぞれの意思決定に役立てるために作成される報告書類であり、その中心となるものが基本財務諸表と呼ばれている損益計算書と貸借対照表である。

財務諸表は、損益計算書、貸借対照表のほかに、企業会計原則で定められている財務諸表付属明細書、利益処分計算書、商法会計で定められている営業報告書、利益処分又は損失の処理に関する議案（利益処分案）等があげられる。

また、上の通常の財務諸表に加えてわが国では証券取引法の間接財務諸表規則により作成される中間損益計算書・中間貸借対照表、連結財務諸表規則により作成される連結損益計算書・連結貸借対照表・連結剰余金計算書が制度化されている。現代の企業は資本での結びつきを基礎に法的に独立した複数の企業が集団を形成し企業経営の効率運営、国際化への道を歩むようになった。日本では、1977年に証券取引法適用会社に対しに示した連結財務諸表の提出が義務づけられている。中間財務諸表は、一年の事業年度毎の報告書では投資者等の利害関係者に十分な情報提供がなされない場合に、会計数値の見積数値も取り入れて有用な情報を早く提供する目的で作成される。

財務諸表分析を目的とする分析のための資料は上にあげた財務諸表が中心になる。財務分析の対象を広げ企業の財務体質・安全性、成長性を総合的にみたり、地域や地球環境に対する取り組みの姿勢はどうであるかとか、単年度等短いスパンの分析のみでなく中・長期にわたる分析を行うとき等には、

---

(3.5) ミドル・ソフトウェア：OSとアプリケーション・プログラムとの間に位置し相互のインターフェースをとるプログラム。

企業の事業目的・事業内容・経営組織・資本構成・株主・株価・配当等々のデータはもとより、製造企業の場合は、生産設備・仕入・生産状況・受注販売状況等々に関する書誌的・数値的データもデータベースの枠組みに入れておく必要がある。

したがって、財務データベースを作成する場合はこれらのあらゆるデータをリストアップしデータ自身の特性と構造を把握するとともに分析手法の側へデータが円滑に受け渡し可能なデータの蓄積構造・ファイル編成方式・データアクセス法が考慮されなければならない。

以上の外、情報ネットワーク環境下の財務データベースに関しては次章（4.4項）で、企業内外各階層の分析者の多様なデータアクセスを想定し、財務諸表を中心とするデータベースのデータ編成、データ蓄積構造、データ操作等、データ構造に視点をおき考察する。

## 第4章 新環境下の財務情報分析システム

### 4.1 ネットワークと新しいマン/マシン・インターフェース

マッキントッシュ社のアイコンを用いた画面インターフェース、Windows (Microsoft 社)、X Window (MIT 開発) 等の台頭によってコンピュータによる情報システムと、人との情報交換の概念・手法が人に近い形の論理で行われるようになり、使用者の経験やコンピュータの知識がなくてもコンピュータが表示し要求する情報を具体的且つ視覚的に判断し捉えることが出来、また、コンピュータに指示することが可能になった。

このコンピュータと人との“ルック&フィール”環境（以下これらの利用環境を総称して“ウインドウズ環境”と呼ぶ）は、コンピュータのスペシャリストと一般ユーザの間の境界をグレーにし、コンピュータに関わりのなかった人、あるいはコンピュータとの距離を感じていた人々とコンピュータとの関係を密にし多くの人々の情報システムへの関心を高めることが出来た。

このようにウインドウズ環境は、コンピュータの一般化・大衆化を促したほか、コンピュータのソフトウェアのオープン化にも大きな影響をあたえている。

従来アプリケーション・プログラムは、特定のコンピュータの機種、OS用に個々につくられていてそのハードウェアとソフトウェアの限定された環境の下でのみ使うことが出来た。したがって、企業のように機種の異なるコンピュータを導入しているところでは機種毎に同じアプリケーション・プログラムを揃えておかなければならないし、そのソフトウェアに関連する付加価値製品を含めて補助的ソフトウェアもそれぞれ必要としていた。

ウインドウズ環境の導入されたコンピュータではアプリケーションを機種

個々に持つ必要はなくウィンドウズ環境用 (for Windows 等) のそれを一つ持つのみでよい。

さらに、ウィンドウズ環境はソフトウェア製品が固有に持つファイル・フォーマットを相互に変換するツール (例えば、Excel / Rotus1・2・3 ファイル変換) を通常備えているため、使用者がそれらのプログラム (ユーティリティ) を探し調達することも必要でなくなった。

このようにウィンドウズ環境は、パーソナル・コンピュータ (PC) の機種に依存することなくあらゆるプラットフォームで、また、アプリケーション・プログラム間の境界を取り払い、コンピュータ使用者がプログラムを生産し、あるいは、プログラムを利用するための思考活動に専念できる場 (開発・利用環境) をつくりあげたといえよう。

情報システム全体の視野で現在のウィンドウズ環境を定義づけるとすれば「コンピュータのソフトウェアの開発、および利用のためのクリアリング・サービス (Clearing Service) を提供し、人と機械との論理ギャップの変換・翻訳を含め快適な使用感を与えるローカルな作業環境」とでもいえよう。

現在情報システムに関して最もよく使われているキーワードは“ネットワーク”であろう。クライアント／サーバー・システム (CSS: Client Server System)、オープン・システム、DCE (分散コンピューティング環境) 等のネットワークのキーコンセプトとなっている用語から個別企業が所有する商標迄、ネットワークに関する新しい用語がつけられ使われてきている。

CSSは、分散配置されたコンピュータの利用環境に於いて情報処理を要求するクライアント・プログラム (依頼人) とサーバー・プログラム (給仕人) が協調して処理を行っていくシステムでネットワークという広域の利用環境の下で行われる情報処理のコンセプトをいう。

オープン・システムは、ソフトウェアの仕様と内容を公開し、分散システム環境でネットワークの設備・装置・機器、および関連するソフトウェアを

一定の基準・規格で相互に接続するシステムをつくることをいう。

DCEは、CSS、オープン・システムとも関連した用語で複数のターミナルから構成されている接続環境を統一したシステムとして捉える環境を云う。

CSS、オープン・システム、DCE等のネットワークに関するコンセプトは、データベースシステムの利用者の側面のみで見た場合膨大なデータと分析プログラム（仮想データベース）が自分の使用している目の前のコンピュータの中にあり、ネットワークを意識することなく独立して個別のコンピュータサービスが受けられるシステムのことをいう。先に触れたウィンドウズ環境の例で言えば、現在のローカルなウィンドウズ環境が論理的にグローバルな範囲にまで拡張されたものとみることが出来る。

現に、インターネット<sup>(4.1)</sup>の世界では通信相手のメール・アドレスを指定するのみで会話的スピードでメール交換が可能であり、インターネット・アドレス、またはドメイン名を用いて世界のデータベース、画像情報へも容易にアプローチすることが出来るようになった。

マン/マシーン・インターフェースが考慮された利用環境<sup>(4.2)</sup>の下であ

---

(4.1) インターネット：世界の150以上の国や地域で、40,000以上のネットワークの相互接続、400万台を超えるホスト・コンピュータが接続され、4,000万人以上の利用者があるといわれている（石田晴久著、「インターネットの現状と課題」HAS研ニュース、No.11、HITAC アカデミックシステム研究会、1995.8.20号、pp.47-54）。

(4.2) 利用者が望んでいる機械との対話環境は、欲している情報、分析ツール（プログラム）が何処にあり、どういう経路でその情報を求めるかではなく、必要とする情報の要求仕様をコンピュータ（情報システム）に指示することで利用者とサービスシステムのインターフェースが確立されその関係の下で所要の情報が獲得出来る仕組みである。扱う情報量が膨大になりそれによってデータ管理構造が複雑になったとき、その部分はシステムが吸収すべきで利用者に新たな知識と時間を要求しないシステムをつくるということが肝要である。

また、研究者、システム開発者（生産者）、システム利用者（受益者）等それぞれの立場で最も能率よく情報処理の行える利用環境を利用階層毎につくることも大切である。

れば、自分の使うデータの所在とその使用法に接近することのみでネットワークの知識を持っている必要はないし、それを意識して使う必要もない。利用による負担金の精算・決済も国際的に流通しているカードで行うことが出来る。

第4章では、第3章で触れた財務データの特殊性と財務分析で予想される利用態様をイメージし、新しいコンピュータの技術とネットワーク環境を組み入れた財務情報分析システムについて考察する。

ネットワークにおける財務データベースの構成の例を 4.4.1 図に示す。

ここで扱う財務データベースは、国内外の各種財務諸表を中心に財務（分析）に関連する基本データを収録しその管理とサービスを提供する“基幹データベース”，および、分析サイドの諸要求に応えられる機能を持った“目的データベース（以下財務データベース）”より成り、各々は役割・分担を異にし論理的には階層で構成される。

基幹データベースのデータ編成は、実績と処理効率の高い“階層構造”，オープン性（仕様公開・互換）の高い“関係型”等が選択される。

データベースの定義・管理用プロシージャ，データベースへのアクセス，変換ツールが整備されていれば“V S A M（仮想記憶アクセス法）”を採用することによって処理要求に対して柔軟性と処理効率の高いシステムを構築することも出来る。

基幹データベースのシステム提供者としての要件は、国際規格，業界標準に準拠したインターフェースで継続利用に応えることが必要である。また、基幹データベースは、高トラフィックになるという特性をもっておりその対策も考慮しておかなければならない。

基幹データベースへのアクセスは次の二つの目的に分けることが出来る。

- (1) システム開発・管理・維持（保守）に関連するアクセス。
- (2) 提供データベースに関する問合わせ・照会・検索等，エンドユーザからの基本的・一般的利用のためのアクセス。

前者の場合の利用者は、SE（システムエンジニア）等システム開発の専門家でありそのインターフェースはホストのデータ管理機能を中心とするオペレーティング・システム（OS）、およびデータベース・マネジメント・システム（DBMS）機能を用いて行われる。

データのアクセス権限の制御も上の制御システムに用意される機能が用いられ、その管理・保守は提供側システム管理者に委ねられる。

後者の場合は、利用目的を異にする幅広い利用者層からアクセスされるためそのインターフェースとしては多様なメニューが必要になる。

すなわち、データベースの標準アクセス言語である“SQL（Structured Query Language）”，“QBE（Query By Example）”はもとより流通ソフトウェア（Lotus1・2・3, Excel 等）のスプレッド（表計算）とのインターフェース、利用マニュアル、ヘルプ機能等々のノウハウの提供が必要である。

さらに、データベースの一般利用者がそれを利用するときに利用のためのインストール等の準備作業をすることなく端末のセッションを開くのみで速やかにデータベースをアクセスし、基本的な分析を行うことの出来るデータベース操作簡易言語等がシステム提供者側機能として整備されていなければならないであろう。

## 4.2 分散データベース

従来データベースといえば提供するデータを中央の大型コンピュータに集中し、その管理運営から利用者へのサービスまで全てのデータ処理をそのコンピュータに構築するという構成がとられていた。

すなわち、次に示す事項は全て中央で行われていた。

- データベース・システムに用意されるデータベース定義言語によって全てのデータファイルのレコード属性、ファイル、およびレコード相互の関連

等の定義（データベース・スキーマ：database schema と呼ばれている）。

- 蓄積されるデータはコンピュータの中では磁気ディスク等に物理的に記録される。それをプログラムや利用する人が扱い易い論理的情報として理解できるようにするため各種の辞書（プログラムで用いるディレクトリ、および、利用者がデータを容易にアクセス可能にするためにつくられるディクショナリ）等データベースのメタ・データの作成管理を行うシステム。
- 利用者に対して、データベースへの照会言語、分析ツール等システム管理側と利用者間のインターフェースに関する部分までを総合的に処理する“データベース・マネージメント・システム（DBMS）”をデータベースの管理中枢部のみに設ける方式（“集中型データベース”と呼ぶことにする）。

この形態のデータベースの特長は、収録・蓄積されたデータは地理的に一カ所の設置場所（サイト）で集中的に管理されるためデータの更新・維持作業がセンターで単独に行えるので更新データと他の関連データとのマッチングが容易にとれること、データベースのエンハンス（機能拡張）を円滑に行い得ること、利用者へのサービスの一貫性が保てること、データベースのソフトウェア障害に対して対応がしやすいこと等があげられる。

一方、欠点としては、現在のようにデータベースの管理機能とそれを支援するソフトウェア技術の高度化が進み、蓄積データも膨大なものになっていく環境の下でサービス水準を高度に保つシステムは、ソフトウェアの側面からみるとその規模と共に複雑なものとなり経済性が低下する。

すなわち、ソフトウェアに関してはその開発が労働集約的な作業に依存し規模の経済は成り立たず、逆に規模の拡大と共にソフトウェアの信頼性は低下しその開発コストは急激に増加する。中・大型コンピュータのハードウェアのコストパフォーマンスに関してみるとパーソナル・コンピュータ（PC）、ワー

クステーション (WS) に比較して現在では相当の開きがある。<sup>(4. 3)</sup>

データベースの収録情報量の増加, データベース管理システムの巨大化, 高いサービス水準の確保, 経済的運営等データベースをとりまく諸課題に解答を与える情報システムとして“分散データベース”が注目されるようになった。

一般にシステムを分散することによって次の事項に関連する効果が期待できる(但し, 分散システムを導入する最も大きな理由は高い経済性といわれているが本叢書の目的とするテーマではないので詳細には触れない)。

- (a) 負荷の分散
- (b) 機能の分散
- (c) 多重(複数)構成による信頼性の向上
- (d) 分散配置(複数サイト)によってもたらされる転送情報量の逓減
- (e) マンパワー, 情報資源の獲得

(a) は, 同時利用の多重度が高く単独の処理装置(CPU)ではユーザの満足する応答時間を確保できない場合に複数サイトに処理装置を分散設置し応答時間の改善を計る。負荷分散が必要となるケースは, 現時点のハードウェア技術では処理要求を満たす処理装置が存在しない場合, すなわち, システム設計書のニーズがハードウェアの技術水準(単独の処理装置でシステムを構成するとして)を超えている場合である。また, 分散システムを構成することによって経済性が確保出来る場合が主要な理由になる。

(b) の機能分散では, いろいろな分け方が考えられる。情報システムのキーコンセプトとして現在最も論じられ, 且つ実用システムとしてシステム開発が続けられているクライアント/サーバーシステム(C/Sシステムと呼ばれている)は機能分散システムの典型である。

---

(4.3) MIPS (Million Instructions per Second) 単価の比較で, メインフレームが \$100,000 であるのに対し, UNIXシステムでは \$1,000 というデータがある。UniForum 社刊, 月刊誌, 1993年4月号。

データベースの収録データの更新・維持を行う管理システムと利用者のデータ検索や分析を支援するサービス・システムを分けること、さらに、サービス・システムを細分化し“ファイル・サーバ”，“CPUサーバ”，“プリント・サーバ”等の提供者側のサービス機能の分散と利用者機能（クライアント）に分け、それぞれのサブシステム相互をネットワークで接続し全体を統括するサーバ（グローバル・サーバ）より構成されるシステムを構築することにより制御の単純化を進めることができる。

（c）通常信頼性概念は、定められた条件・期間の中で目的とするシステムが要求された機能を安定的に実行（稼動）出来る性質を表し、非修理系では確率変数の“故障寿命（MTTF）”，修理が可能な修理系では“平均故障間隔（MTBF）”，“保全性”等で表される。分散データベース・システムの信頼性を、ハードウェアとソフトウェアに分けて考える場合、特に後者に注意を払わなければならない。

分散システムでとくに注意を要するのは、原データが何物であるかを記述するメタデータの整備と蓄積データ相互の関連を明確にしておくことである。たとえば、財務データベースの場合、扱うデータは事象（企業活動）のいろいろな側面の識別可能な特徴を測定し数値の並びで表したものであり、検索したデータが妥当なデータであるか否かは他の資料と照合しない限り解らないことも多く、その意味から分散システムにおけるデータの扱いは厳密を期さなければならない。

自然科学で用いるデータは定義された概念と尺度がはっきりしているので判断しやすいが、人文・社会科学の数値データの場合は事象の概念範囲が広く、また、文脈で判断することは困難なケースも多くデータの並びをみて正しいデータであるか否かは何らかのフィルター（照合、クロスチェック等のチェック・システム等）を通さない限り確認することが出来ない。

検索したデータを単に表として出力する場合は、それが間違ったデータで

あるかどうかは利用者の知識によって確認することが出来るが、検索したデータを何らかの分析ツールの入力データとして用いた場合は、そのデータが妥当なものであったかどうかは分析結果からは断定することは出来ないという危険を内包する。

以上のように、データベースのサービス・システムに於ける信頼性は重要な意味を持つ。

つぎに、分散システムが集中システムに対して信頼性の面で優位となる側面の理由を上げる。

一つは、C/Sシステムで説明したように情報システムの構成を機能毎のサブシステムに分けることによりデータ処理の仕組みが単純化される。これによって信頼性の向上と、システムの点検・整備が容易となりシステムの安全性も上げることが出来る。

二つ目は、分散システムは設備を含めた構築コストを集中システムより低くすることが出来るとされているためバックアップ系等複数システム構成をとることが経済的に可能になる。これによってハードウェアの信頼度を上げることが出来る。

(d) 物資の輸送に於いて需要に対して遅延なく、且つコストの最適化を図るには、ブランチ、工場等の拠点を如何に配置し、如何なる輸送手段・輸送経路・輸送頻度等でスケジュールするかが重要になる。

情報システムの拠点配置に関しても同様のことが当てはまり、情報量のトラフィックを勘案してサイトを配置することにより転送情報量を少なくし通信コストを逡減することが出来る。このことは利用者への応答時間の短縮にもなる。

(e) は、分散システムの特長というよりか国際化の環境下での将来の課題として上げておくことにしたい。

分散システムの考え方は古くからありオペレーティング・システム（OS）でもその発展史をみれば明かなようにOSの重要な課題の一つとして取り組まれてきた。オペレーティング・システムの初期の目的はコンピュータの処理能率を上げることでありそれを実現するために分散処理の概念が取り入れられた。すなわち、一つにはマニュアルな操作を制御言語を用いて仕事をスケジュールしジョブの実行能率を上げること、二つ目は低速の入出力装置（物理速度）と中央処理装置（電子速度）相互の処理能力のアンバランスを解消することから誕生した。

1960年頃IBM-7090システム用に開発された“IOCS(Input Output Control System)”は、中央処理装置とは別に衛星計算機を設け低速の入力装置から読み取ったデータを磁気テープに蓄えること。

また、中央処理装置で処理され磁気テープに入れられた処理結果を低速のプリンタ等の出力装置に出力する働きをもたせた制御プログラムである。

この入出力専用計算機と入出力制御プログラムIOCSで構成するシステムはコンピュータを構成する各装置の処理速度の不均衡を改善するために考案された分散処理システムである。

オペレーティング・システムの考え方が生まれたのは1959年頃につくられたFORTRAN モニタ (FORTRAN II) と言われているのでOSの誕生と同時期に分散システムの実用化がされている。

コンピュータの開発史に大きな変革を与えた“プロジェクトMAC”(MIT)で開発された“MULTICS (Multiplexed Information and Computing Service)”と呼ばれているコンピュータ・ユーティリティでもハードウェア、ソフトウェアに分散システムが取り入れられている。ハードウェアでは、中央処理装置、主記憶装置、入出力制御装置各装置の複数化（多重化）、記憶装置を階層化（主記憶、磁気ドラム、磁気ディスク）し、ソフトウェアでは、記憶階層を用いた仮想記憶方式（Virtual memory）を確立して使用で

きる記憶容量を大幅に拡張した。これによって大規模のプログラムの作成を可能とし、また、多数の利用者の同時利用（TSS）方式の発展を促した。

多重処理、並行処理、並列処理等の分散処理に関連する問題は、分散システムで構成されたシステムをつくる時に解決しなければならないシステム相互の透明性確保、一貫性制御等の機能を持つ分散オペレーティング・システム等、オペレーティング・システムのこれからの主要な課題として研究・開発が続けられてきている。

クライアント／サーバ・システムに代表される分散システムより成る情報システムは、1980年代まで取り組まれ築き上げられてきたメイン・フレーム中心の集中システムと比較しシステム開発に於いて、情報システムの管理・運営に於いても新しいコンセプトの下に時代に即応した根本的な変革が要求される。ハードウェア、および、ソフトウェア両面に於いて中央処理装置を中心に考えられてきた現在までの情報システムの開発のパラダイムを見直すべき時期にきている。

すなわち、ソフトウェアでは従来の集中処理中心の考えで築き上げた情報資産（プログラム）にとらわれることなくスクラップ&ビルドを進め国際規格・業界標準に準拠したオープン性の高い製品指向の情報システムの積み上げ・構築が求められる。

現在のソフトウェアの研究開発、製品の蓄積、流通、標準化の何れの面に於いても新しい情報システムを開発していく環境は成熟しているとみてよい。それは、次に示すことから裏付けることが出来る。

(1) オープン・システムが叫ばれてから久しい。現在は、ネットワークを中心として国際的に標準化が進み規格に準拠した製品がコンピュータ・サプライヤー以外のサード・パーティを含めて競争市場も確立してきた。

これにより、いままで自前のマンパワーで開発していた中間ソフトウェア（関数・サブプログラムを含めて）が費用を別にすれば短期間で求めら

れるようになった。

(2) 分散システムは、UNIXを中心にしたプラットフォームで発展してきている。UNIXは、ソースが公開されているように元からオープンなOSとして発展した基盤ソフトウェアであるということ、その開発思想をユーザも受け入れ機能の拡充を続けてきたという背景がある。

したがって、UNIXユーザならば容易に入手できる“PDS (Public Domain Software)”，フリーウェア，シェアウェア等がシステムに付加され、あるいは、流通システムとして出回り、UNIXで構築するソフトウェアならば比較的短期間で新しいシステムを開発することができるようになった。それらのノウハウの蓄積は膨大であり、またそれを支援する膨大な数のUNIXのエキスパートがいることも見逃すことは出来ない。

### 4.3 分散システムの構造

分散処理は集中処理に対立する概念で、一つの仕事を分散して行わせ全体を統御・遂行させるシステムを分散システムと呼んでいる。

分散システムを情報ネットワーク環境におけるシステム構造の形で類別すると、“垂直分散型”と“水平分散型”に分けることが出来る。

前者は、機能分散、階層構造を特徴としている。前に触れたオペレーティング・システム誕生の元になった“IOCS (入出力制御プログラム)”は垂直分散の一形態といえる。

後者に関しては、現時点ではクライアント／サーバ・システムがその典型であるが、厳密な水平分散システムはなくシステムを細分化したとき下位に位置するサブシステムは垂直分散型のシステムより構成される。

したがって、水平分散システムという場合は情報処理単位が機能で分散されたソフトウェアのシステム構造をいうことにする。

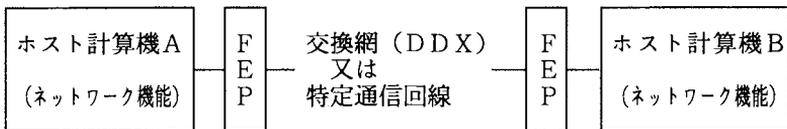
分散システム構築のためには次に示すような要件が設計コンセプトとして求められる。

- (a) 分散システムを構成し制御を行うハードウェア，およびソフトウェア基盤に対して“位置透明性”（トランスペアレンシー）を保障する制御。
  - (b) プロトコルの標準化とマルチベンダーのサポート（オープン性）。
  - (c) ネットワーク環境全体を通じて“ネットワーク透明性”を保障する制御。
- また，分散システムを構築するには，
- (a) 分散システムを記述する言語。
  - (b) 分散システム全体を制御・統括する分散オペレーティング・システム。等の基盤が整備・確立されていなければならない。

#### 《垂直分散システムについて》

垂直分散システムの一例として“N-1 ネットワーク”<sup>(4.4)</sup>を取り上げてみる。

その構成図を次に示す。



- FEP: Front End Processor
- DDX: Denden-Kosha Digital Data Exchange

図は，ホストAとホストBの2つのコンピュータが接続された例であり，複

(4.4) 大学に設置する異機種の研究用計算機を相互に接続するネットワーク。1974年に開発がすすめられ，当初は東京大学—京都大学が接続された。参照資料：HITACマニュアル，「N-1 ネットワークシステム使用の手引き」，マニュアル番号；8090-3-129，日立製作所。

数のシステムが対等の関係で通信網を介して相互に接続されている。N-1 ネット環境に於けるデータ処理は、TSS、および、RJE (Remote Job Entry) を可能とし一つの回線で複数のユーザが相互接続されている指定ホストに対して、これらのデータ処理を並行して行うことが出来る。

ネットワーク・アーキテクチャ (Network Architecture) は、通信回線網を除けば次の三つのプロトコルの階層で構成されている。

- (1) アプリケーション・プロトコル
- (2) HOST/HOSTプロトコル
- (3) FEP/FEPプロトコル

最上位のアプリケーション・プロトコルは、RJE機能と、TSS機能等を行わせるためのプロトコル (NVT : Network Virtual Terminal) を定めている。

RJEプロトコルでは、RJEユーザとRJEサーバに分け、相互間の情報交換を行うためのRJEコマンドを規定し、そのコマンドを用いて入力/出力RJEのジョブの転送要求、ジョブの取消、ジョブの処理状態の問い合わせ、ジョブの終了要求等を行わせる。

NVTプロトコルは、仮想的な標準端末を定義し、サーバ側端末/ユーザ側相互間のデータ通信をネットワークの中で行わせるための通信規約である。

仮想端末は、文字型装置で仮想キーボード、仮想プリンタを持ち、その動作を制御するコードを含む伝送制御コード、および、NVTコマンドを仮想キーボードから発生させる装置である。NVTは、アプリケーションに依存しないプロトコルで、TSSサービスはこのプロトコルを用いて行われる。

HOST/HOSTプロトコルは、ホスト間で交換を行うためのコマンド (リンク確立要求、要求に対する肯定応答、リンクの解放等) を定めNCP (Network Control Program) を介してHOST相互の論理的な通信路の確立 (リンク) ・解放の制御の下で、データ転送を行わせる機能階層である。

F E P / F E P (Front End Processor) プロトコルは、F E P間、あるいは、F E Pとパケット交換網間に論理的なチャネルの確立・解放、データ転送を行わせるための通信回線 (D D X, 特定通信回線) の制御を行う。

N-1 ネットワークのサーバ側、または、ユーザ側のハードウェア構成は、

- (a) 中央処理装置
- (b) 入出力処理装置 (入出力チャネル)
- (c) F E P (前置き処理装置)
- (d) 通信網

また、ソフトウェアのシステム構成でみると、

- (a) アプリケーション (T S S, R J E)
- (b) ネットワーク制御プログラム
- (c) 回線制御プログラム

に分けられる。何れも機能で分けられ、且つ階層構成より成る垂直分散型システムである。

#### 4.4 財務データベース分散システム

図 4.4.1 にプロトコル(T C P / I P等)による通信路を軸とするサーバ / ユーザそれぞれのソフトウェア階層が対称に配置されたデータベースの分散システムの構成を示す。

互いに独立したオープン性の高いソフトウェアを一つの単位とし、それらを階層的に配置することによってまとまった仕事を行う情報システムを構成することができる。このソフトウェア構造を採ることにより、システムの保守性・機能拡張性を高め、また、ソフトウェア市場で評価の高い製品を選択することによってシステムの信頼性が高められると共に、ソフトウェアの市場価格による利益も得られ経済的なシステムを構築することが出来る。

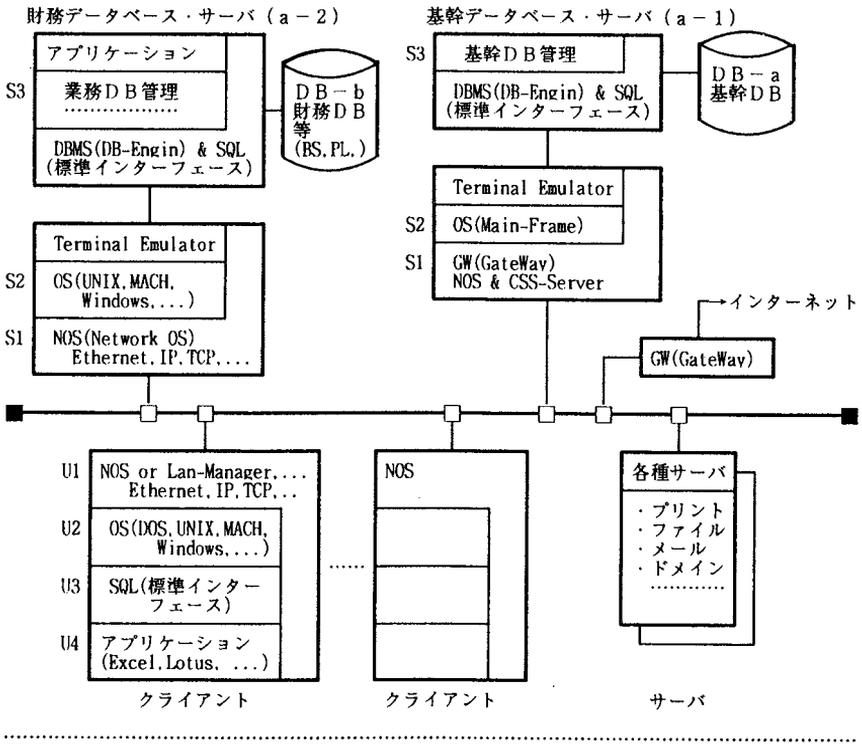


図 4.4.1 財務データベース分散システム (例1)

図に示した分散システムは、メイン・フレーム上に構築された例えばリレーショナル型データベース（RDB型）をネットワーク環境の下で利用する分散システムに拡張したシステムで、メイン・フレームの高速処理・高信頼性のデータベース管理機能と、ワークステーション（WS）、パーソナル・コンピュータ（PC）のネットワーク管理機能、ウインドウズやスプレッド型計算・処理等のユーザ・インターフェースの優れた機能を相互に取り入れたシステムである。

分散システムとして捉えると、まずデータベースを管理・維持しユーザの必要とする情報（利用のノウハウ、データ、ツール等）を提供する“サーバ”，利用者サイトの“ユーザ（クライアント）”に分けられる。

TCP/IP等による通信路を基準にサーバ側のソフトウェア・システムを表すと次の二つのサイトに分けられ、それぞれは階層で構成される。

- (a) サーバ側は、基幹データベースを持つメインフレーム・サイト
- (b) 業務データベース・システムを構成する分散サイト

この構成の下では、基幹データベース、および業務データベース各々のDB管理を行う管理システム（DB-a, および DB-b）は、相互のデータベースを統一的に扱うための制御とインターフェース機能を持つことが要求される。

次の各項目は、説明の便宜上 図 4.4.1と記号を共通にした項目リストである。

《注記》

- ・ [a-n] はサーバ， [b] はユーザ側ソフトウェア階層を表す。
- ・ s n（サーバ側）， u n（ユーザ側）は、各々の階層を表す。  
n は、階層の水準を表し大きい数字が高水準に位置する。
- ・ NOS：“Network Operating System”の略。

## 〔a-1〕 サーバ側ソフトウェア階層 (DB-a)

s 3 : メイン・フレームDB管理

s 2 : OS (メイン・フレーム)

s 1 : ゲートウェイ (NOS機能を持つCSSサーバ)

## 〔a-2〕 サーバ側ソフトウェア階層 (DB-b)

s 3 : 分散DBサーバ (DB-b)

- SQLアクセス

- 財務諸表作成・管理・監査, クライアント管理, 財務分析ツール等

s 2 : OS (UNIX, Mach<sup>(4,5)</sup>等)

s 1 : ネットワークOS (NOS)

## 〔b〕 ユーザ側ソフトウェア階層

u 1 : ネットワークOS (NOS)

u 2 : • DOS and/or Windows

- UNIX and X Window

- その他のOS環境

u 3 : SQL/DBアクセス・マクロ等ミドル・ソフトウェア

u 4 : アプリケーション

(財務分析システム, Excel, Lotus 1・2・3, SQLアクセス等)

分散システムをソフトウェアの階層としてみた場合に最も理想的な形は、それぞれの階層が独立し他の階層（上位、または下位の隣接する階層）に変更があった場合に当該層は影響をうけないこと、自分の階層に変更があっても他に影響を与えない閉じたシステムで構成され且つ上位の階層からの要求に対して

(4.5) Mach : 米国カーネギーメロン大学で開発 (1985年頃) された分散システム、および並列システム用に開発されたオペレーティング・システム。

プロトコルで定められたサービスを提供することである。さらに、通信路等を基軸にしてサーバとユーザ（プラットフォームを異にするユーザを含め）相互が対称型に配置されることが望まれる。

しかし、このようなシステムは設計図として書くことは出来るが現実のシステムとしては、蓄積されてきた膨大なソフトウェア資産の継承の必要性、マルチベンダーより構成されるコンピューティング環境の確立、ネットワークの技術進歩の構築システムへの反映等々の理由により、業務目的に合った固有のネットワーク・トポロジーで構成されているのが現状である。これからの課題としては、標準化・制度化されていないベンダー独自のソフトウェア製品の国際標準・業界標準を一層進めていくこと、UNIX等のシステムを記述する汎用プログラミング言語の規格統一を進めていくことが必要である。

図 4.4.1 に示した、財務データベース分散システムについてそれぞれの階層に位置するサブシステムの機能・構造等に関すること、および、隣接するサブシステム間の関係、対応する階層のサブシステム（ユーザ、またはサーバ）との関連等についてみていくことにする。

#### 4.4.1 サーバ側ソフトウェア階層 (a-1 : DBa)

##### 4.4.1 a メイン・フレームDB管理 (s3)

###### (1) 基幹データベースの役割と位置づけ

データベースは、3.1項「データベースの発展と構築の要件」で触れたようにその原型は、米・英にみられる資料保管所（Data archives）から出発し、1950年代の終わりから1960年代の初めにかけて米軍が開発した管理指令システムがデータベース進展の大きな役割を果たした。

以後、データ検索機能、レポート作成の容易化、データベースの記述・操作

の専用言語の開発等と広がりを見せ、1960年代の後半にはデータベース・システム（データベースとそれを管理するシステムの総称をこの用語で呼ぶことにする）の急激な展開がなされた。

この動きに相前後してデータベース・システムに大きなインパクトを与えたのは、プログラム言語COBOLの開発母体である“CODASYL (The Conference on Data Systems Languages)”システム委員会で、その成果はデータベース・システムのみならず情報システム発展の輝かしい足跡としてその歴史に記録されている。CODASYL型は網モデル（ネットワーク・モデルともいう）と呼ばれるもので、親子集合型より成る二層のレコード階層構造の組み合わせで表されるモデルをいい、データベース・システムでは“IDS (Integrated Data Store: GE社)”, “TOTAL (Cincom Systems, Inc.)”等がこの型に属する。

網モデルの変形として“階層モデル (hierarchical model)”の概念が生まれ次に取り上げる関係モデルと合わせてデータベースのモデルの一角をなす。階層モデルは木型モデルとも呼ばれ、データベース・スキーマ<sup>(4.6)</sup>がレコード型の下に複数個のレコード型を階層的に配置してつくられるモデルである。階層モデル型のデータベース・システムとしては、IBMのGIS (Generalized Information System) およびIMS (Information Management System), MUMPS (Massachusetts General Hospital) がよく知られている。

CODASYLの活動と共にデータベース・システムの発展史に大きな変革をもたらし、システム開発のパラダイムを変えたのは、IBMサンノゼ (San Jose) 研究所のコード (E.F.Codd) の提案した“関係モデル”である。

---

(4.6) データベース・スキーマ：データベースが対象とする世界を知覚する際の基本的な認知構造をいう。概念スキーマ（データベースの持つ意味を一般的且つ忠実に表現する構造）、外部スキーマ（特定の応用分野から論理的視点でデータベースを見る時の構造）、内部スキーマ（データベース記述の物理構造）に分けられる。

関係モデルでは、そのモデルで提案された4つの関係演算（射影：projection，選択：selection，結合：join，除算：division）と、4つの集合演算が用いられる。データ構造として関係の表現に表を利用し、関係演算をデータベースのデータ操作に用いたデータベースを関係データベース・システムといっている。

初期の関係データベース・システムとしては、IBMのサンノゼ研究所で開発されたSystem Rがよく知られている。また、データベースのデータ操作言語である関係データベース言語としては、ALPHA, QBE, SQL等がある。SQLは、1987年に国際規格とJIS規格に制定され、現在関係データベースへの標準インタフェース言語としてよく使われている。

上に示したようにデータベースの理論と実用へのシステム化の歴史は30年以上におよび関連するソフトウェア資産とそのノウハウの蓄積は膨大なものである。

現在、情報システム開発の新しいパラダイムとしてクライアント/サーバ・システムが注目されているが、現在までに蓄積された資産をプログラム変換・データ変換等効果的なツールを用いて新しい情報基盤へ継承していくことが求められる。

データベース構築のプラットフォームのOSにUNIX（或いは、NetWare, Windows-NT等）を備えたコンピュータが選択されるのはこれからのデータベースの方向性としては容認されるが、オープンなシステム環境下でのデータベースの開発・運用管理に必要なシステム・ソフトウェア、ツール群がメインフレームのそれに比較して現在十分ではないこと、その環境の下でのシステム構築も開発途上にあること、さらに重要なことはオープンシステムの新しいパラダイムの下で大規模のシステム（基幹データベース）を開発していける技術者が現時点では少ないといわれている。

このような理由から大規模のデータベースの構築は、信頼性・安定性・処理能力等々で現在のメインフレームが勝っており、基幹データベースの構築は従

来のシステム基盤が活かされるべきであろう。

## (2) 分散システムと基幹データベース

分散処理・分散システムは集中処理・集中システムに対立する概念で情報処理の理論と実践両面にわたって古くから関心が持たれてきた研究開発のテーマである。現在あらゆる情報システムにおいて注目されているオープンシステム概念、クライアント／サーバシステムは分散処理と密接な関係にある。

データベースの分散化が試みられるのは次のような背景による。

### a) データベースのコスト要因

扱うデータ量が膨大でありそのデータ管理を集中化方式で行うことによる情報システムの複雑化、データ処理効率の低下等の要因でデータベースの経済運営が出来なくなる場合。たとえば、集中システムの処理に用いられる大型コンピュータと分散システムに用いられる小型コンピュータの間には、ハードウェア／ソフトウェア両面において開発・運営コストの開きがあることが指摘されている。

### b) データベースの安全性

災害発生時、あるいは、特定のベンダーで構築したシステム等による危険を分散しデータベースの安全を確保する。

また、基幹データベースの全データが有機的に関連づけられ、全機能が分散データベースの総合機能で実現出来るシステムを構築することによって安全性を高めることが出来る。

### c) データベースの拡張・統合化

データベースの統合化は、現在まで個々独立につくられてきた複数のデータベースをデータ構造、データアクセス、データ転送、ユーザインターフェース等々の汎用化・標準化を進め統合データベースを再構築しようとする試みである。

分散データベースシステムへの取り組みは、対象データ領域の広がりと量的

拡大に対してデータ資源をデータの発生する拠点、部門、データのカテゴリー等によって適正に配置していかなければならなくなってきたこと、利用者の求める情報が専門且つ迅速な応答を必要としてきたため広範囲のニーズに応えるメニューを用意し、信頼のおける高い水準のサービスを重点的に継続して提供していくことがシステム開発側に求められてきたことによる。

システム開発・運営面に於いて高いコストパフォーマンスで、機能拡張が長期に渡って円滑に進められる分散システムの構築がオープンシステム基盤の上で可能になったということも分散システム進展への背景になっているといえよう。

コンピュータのハードウェア、およびソフトウェアの分散システムないし並列処理への試みは、コンピュータに学習機能を持たせ知識情報システムとして人の知的活動のアシストを機械に行わせようとする研究・開発の問題とならび、コンピュータが誕生して以来その時代その時点の将来の研究課題として、また実用システム化の問題として多くのエネルギーがつき込まれてきた。現在その研究成果および実用化はめざましいものがある。

並列処理に関しては、64台の演算装置を格子状に結合した構造を持つ並列計算機“Illiac IV”が1970年頃イリノイ大学で完成されたのに始まり、パイプライン制御計算機、多重プロセッサ、データフロー計算機等が開発されている。現在では、多数のマイクロプロセッサを並列に結合したMPP（Massively Parallel Processor）が各社で開発され商用として使用されるようになった。しかし、まだ適用されている分野は科学技術計算に使われているのが殆ど（1992年時点で98%、Gartner Group）であり、本格的なMPPの採用へはハードウェアの技術進歩のみならず、並列システムの開発・運用管理の各種ツール、MPP用データベース管理システム等、システム全体の統合管理を行うOS（UNIXが多い）を含め標準化されているソフトウェアは未だ少ない。

分散システムに関する研究開発はかなり古くから取り組まれてきている。初

期の端末装置はキーボードによるデータ入力を中央処理装置が受け入れ、中央処理装置から送られてくるデータ（文字列・特殊文字・エスケープシーケンス等）を受信し文字パターンを生成して画面に表示させるという単機能の装置であったが、端末装置にパーソナルコンピュータが採用されるようになって端末のインテリジェント化が進み、MML（Micro-Mainframe Link）と呼ばれるパーソナル（マイクロ）とメインフレームコンピュータ相互を接続する新たな接続環境が考えられ注目されるようになった。

従来メインフレームが持っていた端末機能を、端末として機能するパーソナルコンピュータ側に持たせ（端末エミュレータ）メインフレームの負荷を端末に分散させる機能、メインフレームと端末側のパーソナルコンピュータ間の境界に各々データ交換の制御プログラムを走らせ相互間のファイル転送を行わせる機能、メインフレームのプログラミング言語の記述文法と同じプログラミング言語仕様を受け入れる言語プロセッサをパーソナルコンピュータ側に用意し従来メインフレームで行っていた処理を端末のコンピュータに行わせる等はMML概念の所期の成果である。MML環境はデータベース管理システムの分散システム機能の研究開発を促し現在の分散システム化への関心と道を拓いた概念といえよう。

分散システム化の必要性は、現在情報システムが抱えているバックログの解消、時代のニーズを情報システムに取り入れていかなければならないこと、またコストパフォーマンスという命題等情報システム構築のインフラストラクチャに関わる問題であり目標でもある。それ故に分散システム構築は困難を伴い未解決・未開発の多くの課題が残されている。

分散データベースシステムの構築には、次のようなシステムコンセプト、データベース管理システム（DBMS）、およびネットワークに関する基盤ソフトウェアの整備が必要条件となる。

分散配置されたデータを統合しユーザに対して一貫した共通のインターフェ

ースを提供していくときに求められる“データの透明性”の保証，データ分散化に伴うデータの開放性，データ移動等ネットワーク環境でのデータ態様に対して求められる“データの安全性”，“データのセキュリティ”等が分散データベースの基本的な要件としてシステム設計・システム管理面に求められる。

分散システムを構築する基盤ソフトウェアでは，ネットワークを制御するものとデータベースを管理・運営するソフトウェアに分けられる。

後者は，またデータ定義・記述・操作を行うDBMSと操作ツール（ユーザインターフェース）に分けられる。

ネットワーク技術は，分散システムの構築基盤でありその技術の発展はクライアント／サーバー・システムのコンセプトを現実のものとし，分散システムの研究開発への関心を高めてきた。

我国は1970年代の初めまでコンピュータと通信回線を接続したシステムの業務利用は公衆電気通信法で禁止されていたため，データ通信網の構築は専用回線によるポイントツープointの形態で企業あるいは企業グループ独自のいわゆる閉じたネットワークがつくられていた。1972年と1982年の2回にわたる同法規の改正により通信相手によって課せられていた制限が段階的に緩和され，1985年度に公衆電気通信法が廃止され，新たに電気通信事業法が施行され日本電信電話（NTT）以外でも通信事業を目的とする企業を設立することが出来るようになった。

このデータ通信の自由化は，コンピュータを用いる情報システムの構築に革新的な変革をもたらした。営利を目的とした通信事業者も名乗りを上げ競争市場が確立されたことによりデータ通信のみならずネットワーク技術の進歩と技術の多様な広がりを見た。ネットワークの管理，また利用者にとっては低コストで高速且つ高品質のデータ通信のサービスが受けられるという利益から市場も活性化され拡大し今日に至っている。

ここで，ネットワークの現在，および将来の展望を含めて重要なインフラ

トラクチャを上げてみる。データ通信の基盤に関わることでは、公衆電気通信網の“ISDN”，広域データ通信網では“インターネット”が上げられる。

(a) ISDN (Integrated Services Digital Network)

ISDNは従来のアナログの電話網をデジタル化し高速・多様なデータ通信に応えられるサービスとして1980年代の終わりにCCITT (国際電信電話諮問委員会)で勧告され、従来公衆電気通信網ではアナログ回線を用いた電話系サービスで電話またはデータ通信の何れかしか行えなかったが、ISDNでは一つの回線(1加入単位)で複数のデジタルチャンネルがバインドされ、それぞれのチャンネルは回線交換、パケット交換、信号等の通信モードの設定、およびマルチメディア(音声・データ・画像等)データの通信も可能であり、用途・目的に応じて回線交換、パケット交換(パケット単位の蓄積交換)何れの方式でも行うことが出来る。

我国では、NTTのINSネット64(1988年4月サービス開始)、INSネット1500(1989年6月開始)の名称で、国際通信では、KDDが1989年6月からISDNサービスを行っている。

(b) インターネットは、米国国防省高等研究計画局(DARPA: Defense Advanced Research Projects Agency)の後援のもとに開発された“ARPAnet”が基礎になり地球規模のコンピュータ相互接続のネットワークに成長した。

ARPAの開発した“TCP/IP”インターネット・プロトコルは、コンピュータ相互接続の通信規約として国際的に支持され現在のネットワークシステムのほとんどはこの方式を採用している。

インターネットを介した情報利用システム<sup>(4.7)</sup>は米国を中心に開発されそ

---

(4.7) インターネットを用いたサービスにWWW(World Wide Web)がある。米国NCSA(National Center for Supercomputing Applications)の開発した“Mosaic”は人気の高いシステムの一つで研究情報や企業情報等を国際的な規模で流通(発受信)させることが出来る。

の利用の反響は、調査研究分野では研究・開発の方法（研究実績の調査、データ利用、共同研究等）、ビジネスの分野では情報伝達・利用の方法（場所と時間の観念、機会利用、相互通信等）に関して従来おこなってきた方法の再点検をしなければならないような関心の高まりである。

次に、分散システムのネットワークのトポロジー（構成・形状）に関してはネットワークの理解を深めることはもとよりシステム設計の基本問題でもあるのでその構造面に関して触れることにする。

情報ネットワークの基盤づくりのための法・規則・標準の制定、通信網の整備、ネットワーク技術の発展と広がりによって、分散型データベース構築のための諸々のニーズを満たすことが可能になった。たとえば、従来(1980年以前)のように開発者自身がネットワークを構成する独自の物品・設備・ソフトウェアを調達するのではなく業界標準の製品（TCP/IPプロトコル等）や国際標準に準拠した製品・設備を用いてオープンなネットワーク環境を短期間でつくり上げることが出来るようになった。

しかし、オープン環境の下に構築されるネットワークは、それぞれの役割を持った設備・装置群とそれを駆動させる多種多様のソフトウェア群がネットワークの空間に配置され益々複雑な構成になり全体を一望することが難しくなってきた。

ここでは、ネットワークとユーザ（端末）との関わりを把握するため、面で構成されるネットワークのトポロジーを幾つかの系統（線形）で捉えユーザとネットワークの関係を見ていくことにする。

すなわち、LAN<sup>(4.8)</sup>を基点にしそれに接続される各種の情報資源（ネットワーク機器、通信メディア、プラットフォーム等）の広がりとして捉えたとネットワークとユーザの関係をシンプルな構造で見ることが出来る。

---

(4.8) LAN (Local Area Network) : 建物の単位等狭い範囲内に敷設されたネットワーク。Ethernet型とToken ring型に分けられる。

## (a) LAN—Client (または, Server)

イーサネット、トークンリングはそれぞれLANの物理層およびデータリンク層のプロトコルとして国際標準化 (IEEE 802.3 および IEEE 802.4) <sup>(4.9)</sup> が行われており、端末 (コンピュータ) 相互を結び付けるソフトウェアもTCP/IP等オープン環境で動作する標準ソフトを容易に調達することが出来る。

クライアントとしては、通常仮想端末 (Virtual Terminal) サービス機能 (Telnet) とファイル転送機能 (FTP) を備えたコンピュータが用いられる。<sup>(4.10)</sup>

仮想端末は、相手のリモート・ホストからは同一のインターフェースに見える仕様を持つ端末をいい、PC・WS等コンピュータの型に依存することなくアプリケーションに合った装置を選択することができる。

さらに、複数のサーバと同時に通信 (データ処理) を行い複数のウィンドウを同一画面に表示する機能、たとえば、MITで開発されたUNIXの下で稼働する“Xウィンドウ”のような環境を整備していることが望まれる。

Xウィンドウは、クライアントとサーバが連携しマルチタスクで実行される複数のプロセスそれぞれの動作状況を一つの画面に表示させるシステムである。Xウィンドウは通常のクライアント/サーバシステムと異なり利用者が操作している端末がサーバとなって画面制御の標準的な機能をクライアントに提供する。クライアントはXサーバに対してウィンドウ (ターミナルウィンドウ) を開くサービスを依頼することによりアプリケーションを実行する環境をXサーバの画面上につくる。ネットワークにあるリモート・ログイン可能なホストはクライアントになり得るので複数のクライアントによってつくられるターミナルウィンドウをXサーバの画面に集約しそれぞれの仕事を並行的にすすめるこ

---

(4.9) IEEE 802 規格は、アメリカの学会である IEEE (Institute of Electrical and Electronics Engineers) の委員会で作成され I S O に提案されそれを受けて I S O が標準規格として制定したものである。

(4.10) TCP/IP (Transmission Control Protocol/Internet Protocol) プロトコルは Telnet と FTP の機能を標準で持つ。

とが出来る。

次にネットワーク環境におけるサーバとしては、ファイルの定義・蓄積・更新・保全・セキュリティー・検索等を集中的に制御する“ファイルサーバ”，プリントに関するあらゆるサービス（文字フォントの整備・各種プリントモードのエミュレーション・書式制御等）を行う“プリントサーバ”，ネーム（ドメイン）サーバ等が考えられる。

(b) LAN—ルータ<sup>(4.11)</sup>，または，LAN—ブリッジ<sup>(4.12)</sup>

LAN—ルーター—FDDI<sup>(4.13)</sup>

LANの構成単位はシステムの経済運営等全体の管理・運営の方針，トラフィックの確保，LAN設備の規格・技術的な制約等から最適の規模と配置が定められる。技術仕様に関しては、イーサネットLANの場合一つのセグメントの敷設の総延長制限（500メートル以内），セグメント数等の物理的制限があり，また転送速度の上限が定められているためLAN（サブネット）を大規模の構成にすることは出来ない。

現在はLAN相互を接続する装置としてルータが最もよく用いられている。

ルータは，OSI<sup>(4.14)</sup>ネットワーク7層モデルのネットワーク層（レイヤ3）

- 
- (4.11) ルータ (router) : OSIネットワークモデルのネットワーク層（3層）のセグメントを結合する装置。通常複数のLANの接続，インターネット・プロトコル（IP）パケットの経路決定，配置等の制御を行う。
- (4.12) ブリッジ (bridge) : OSIネットワークモデルのデータリンク層（2層）の制御を行う装置。通常物理的に異なるLANを結合する働きを持つ。
- (4.13) FDDI : Fiber Digital Data Interface の略。光ファイバをループ状に敷設しLANを結合するための標準化プロトコル。
- (4.14) OSI (Open Systems Interconnection) ネットワークモデル: ネットワーク・アーキテクチャに関する基本参照モデル。  
物理（1層），データ・リンク（2層），ネットワーク（3層），トランスポート（4層），セッション（5層），プレゼンテーション（6層），アプリケーション（7層）に分けられる。

に位置しネットワーク内を走るパケットの経路決定（ルーティング）等ネットワークセグメント（プロトコル、発信・発着アドレス、データ等を持ったデータブロック）の制御・結合を行う。最近では、ルータは高性能・高機能化がすすみマルチプロトコル（TCP/IP、OSI、AppleTalk、IPX等）をサポートするもの、ネットワークのバックボーンとして用いられるFDDIへ接続できる製品がつくられている。

(c) LAN—ゲートウェイ<sup>(4.15)</sup>

ルータによって接続されるLAN環境は、自分のLANと接続相手のLAN環境（プロトコル）が同じであるか、あるいはルータがサポートする標準規格ないし業界標準のプロトコルでそれぞれが稼働されているLANでなければオープン環境で利用していくことは出来ない。

相互のLAN環境が異なる場合、例えば大型メインフレームで用いられていた通信制御・端末制御の環境をオープンなLANに直接組み入れることはできない。LANに接続された端末からメインフレームに構築されたシステムを利用していくにはメインフレームとLANの間にゲートウェイを設けプロトコルの変換をそこで行う必要がある。ISDN等通信基盤の異なる通信網と接続する場合も同様である。

(d) WAN<sup>(4.16)</sup>—GW—FDDI—ルータ—LAN—クライアント  
(または、サーバ)

最も標準的なネットワークのトポロジー（構成・形状）をWAN側から見た階層である。

---

(4.15) ゲートウェイ (GW: Gateway) : OSI参照モデルのネットワーク層以上のプロトコルを変換する装置。

(4.16) WAN (Wide Area Network) : 広域ネットワーク。

今後は、バックボーンの超高速転送、マルチメディアへの対応というニーズから次世代の情報通信基盤として注目されている“ATM”が取り入れられ、次のような系統になっていくものと予想される。最終的にはデータ通信網も現在の電話網のように国際的に端末が相互接続されWANとLANの境界がなくなっていくと予想される。

WAN—ATM—LAN—クライアント（または、サーバ）

ATMは、現在バックボーン環境を形成している“FDDI／ルータ”の基盤の機能を代替するもので、LANの接続制御をFDDI／ルータがソフトウェアで行っていたものをハードウェア主体の制御にしたシステムである。データの管理単位（53バイト）を小さくして低速から超高速（トータルで数ギガバイト）までトラフィックの幅を持ち将来のマルチメディアにも対応出来る通信基盤として注目を集めている。

《記》

ATM (Asynchronous Transfer Mode) : 非同期転送モード。

ATM技術の特徴としては次のような事項が上げられている。<sup>(4.17)</sup>

- マルチメディア通信（データ、音声、動画等）に最適で効率の良い大容量リアルタイム通信が実現出来る。
- QOS (Quality of Service) 制御を可能にする。
- 運用・管理コストの低減、柔軟性、拡張性に富む。
- 伝送媒体としては、光ファイバ、ツイストペア・ケーブルが用いられる。

### (3) 財務基幹データベース

財務基幹データベースの役割は、分散配置される部門データベース（分散サイト）に対してどのようなデータを供給し如何なるサービスを提供するかとい

---

(4.17)「情報スーパーハイウェイ時代にこたえるATM LAN」, 西島富久他, 日立評論, Vol.76 (1994/11)

うことと、その運用・管理とサービスに最も大きな影響を与えるデータベース管理システム（DBMS：Database Management System）に何を選択するかが重要な課題となる。<sup>(4.18)</sup>

### （3 a）収録データとデータ提供支援サービス

財務基幹データベースに収録されるデータ対象は、分散配置される部門データベースに供給する財務諸表等の数値データ、およびデータ構造を記述するメタデータ（例えばデータレコードの名前、データフィールドに関するデータ形式・データサイズ等データそのものを記述する辞書）が基本データとして上げられる。

書誌的データとしては、データの作成編集更新記録、出所・管理管轄部門（機関）等データの所在に関する情報、データ利用上の注意・注釈等の非数値情報等である。

本稿で扱う財務データベースは、機能分散型を想定しているためそのデータベースはそれぞれの分散サイト（財務分析サーバ、財務諸表作成管理サーバ、資本・予算・原価等の管理サーバ等）で要求される原データ、およびそのメタデータを供給していただける能力が要請される。

また、分散サイトを含めた全システムの管理運営支援機能として重要なことは、収録全データを対象としたデータ検索・利用システムの提供、データ編集操作を行う各種ユーティリティ、ツールの提供、さらに分散サイトで発生したデータの更新情報、システム改善要求等の情報を収集・整理し調整すること、各サイトに対して整合性の保たれたデータ（データの透明性確保）の提供が出

---

(4.18) 我々が構築した“国際経済経営データベース”は、データ管理にV S A M（仮想記憶アクセス法。VSAM: Virtual Storage Access Method）を採用し、データのあらゆる制御・操作をコマンド・プロシージャ（TSS コマンド）によって行っている。この方法ではツール類を全て自前で用意しなければならないが汎用性・拡張性・柔軟性の高いシステムを構築することが出来る。

来る運用管理システムを基幹データベースは備えている必要がある。

### (3 b) DBMSと操作言語（照会言語）

計画されたデータベースの仕様を満たし高いサービスレベルで効率的に運用管理を行っていかれるかどうかはその作成支援システムであるDBMSの機能に依存する。データベース・システムの開発、すなわち運用管理システム、ユーザへのシステム開発支援ツール、ユーザインターフェース等全ての面でDBMS自身を持つ記述言語、検索・照会言語等の言語仕様（文法）に従って記述しなければならないこと、また近来はネットワークに対応しオープンなシステム環境に適応する柔軟性の高いシステムで且つ標準に準拠するインターフェースを備えたDBMSが要求される。

DBMSは、データ構造、データ編成の面から、階層型（ツリー）、網型（ネットワーク）、関係型（リレーショナル）に分けられるが、ネットワーク環境に於ける財務データベース構築には次の理由により関係型のDBMS（RDBMS:Relational ---）が選択される場合が多い。

- データの構造を物理構造から開放し論理構造のみで捉えることが出来る。  
これによってデータの独立性が保たれデータアクセスに関連するプログラムの記述が容易になりユーザ自身の手によるシステム開発の比重を高めることが出来る。
- データの管理単位は表であり財務諸表等現実のデータとコンピュータで扱うデータと対応付けて処理をする事が出来る。階層型、網型では表をレコード集合で表し、そのアクセスの手順も指定しなければならない。
- データの照会言語として国際的に標準化されている非手続き言語の“SQL (Structured Query Language)”を大型・小型コンピュータの区別なく広範囲のレンジで利用することが出来るため、メインフレームに構築した基幹データベースと小型コンピュータを用いた分散サイトのデータベ

スとの親和性を高めることが出来る。

### (3c) 財務データバンクの構築

財務諸表を中心とする財務関連のデータを体系的に整備・蓄積し、その運用・管理を行う“財務データバンク”は、財務情報分析システムの構成の中で重要な位置を占める。

財務諸表は、貸借対照表・損益計算書・（製造原価報告書）・利益処分計算書・各種財務諸表明細表等があり業種（製造業、金融・保険、運輸、エネルギー、サービス等）によって分類・整理方法が異なる。決算形態も本決算・中間決算・連結決算に分けられ、また、国によって会計規則・制度・原則が異なる。これらのデータを統一したデータ構造で収録することは多くの困難を伴う。利用法に関しても国による会計制度・分類基準等の特殊性によりデータ変換を必要とする場合もある。

この状況下で財務データのシステム化を進めるには対象領域を統一的に管理できる単位まで下げ、そこで生じた問題を整理し逐次対象を広げ次のシステムに、蓄積したノウハウを生かしていくシステムズ・アプローチは現実的である。

#### 4.4.1 b 財務情報分析とSECRETARY

財務データベースを公開し利用に供するには、システムの管理・維持業務のほか、利用形態を特定できない各階層の利用者に対して、収録データの探索・検索・表示機能、データ分析・解析機能等想定される諸々のサービスを提供するサーバ機能が必要である。我々が開発した分析言語を持つ経済経営分野を対象とする情報分析システム（SECRETARY）は、ネットワークの配置の中ではこの部分のサーバとして位置づけられる。同情報分析システムの設計理念を次に示す。

## (a) 問題を記述することの出来るシステム

分析ツールには、二つの相反する便宜性が共に求められる。

第一は、突発的・非定型・緊急利用等の場合に容易且つ速やかに解答を与える問題解決型のサービス。分析メニューを提示（例示）し利用者が選択する方法、利用者固有の分析アルゴリズムを登録しそのカタログ名と制御情報を指定することで分析を行わせる方法等が考えられる。

第二は、自由度の高い分析記述能力（機能）を備えたシステム。

問題解決型の分析システムは、システム自身の持つ分析対象に対してはプログラム化された結果情報を効率且つ速やかに出力するがその手法を越えて新たな分析論理の展開を同システムを用いて行うことは出来ない。情報分析システムに問題記述能力を持たせることにより分析対象に対してより高い適応性と柔軟性を持った分析システムをつくることができる。

## (b) 分析者の知的活動（学習）の環境を確立するシステム

問題解決の意思決定過程では、問題（現象・仕組み・現象間の関連等）を詳細且つ的確に表現（記述）する言語を持ち、その言語による分析の経過が記録（記憶）され、記録された論理は分析者の思考活動の速度（実時間）で呼出・再現が可能であり、実験の初期設定なり条件を変えて試行／評価が連続して行える知的活動の環境を備えた制御システムをつくる。

## (c) 大規模のデータベースを構築し分析システムと有機的に結合させる

経済・経営分野における調査・分析・研究ではデータが重要な意味を持つ。与えられた課題に対して解答が得られるまでの時間、費やすエネルギーは、データ探索・整理等データの扱いに関連する作業が高い比重を占める。

したがって、データ処理の能率化を図るには関連データを体系的に整備したデータベースを構築することと、情報分析システムとデータベースが有機的に結合されている必要がある。

## (d) データベースに関する情報提供と利用支援システムを備えること

データベースは、組織的に収集されたデータの集合であるが構築によって静的に収まるものではなく動的なもので利用ニーズに応じて常にデータの更新・追加等が行われる。したがって、データベースの提供者は収録データとサービス・メニューの現状に関して、広報的なサービスを行うことと利用者の問い合わせに応じていくシステムを持つ必要がある。

上の (a) ・ (b) 二点の問題は、目的言語である情報分析用言語の設計問題に帰着し、その分析言語の体系化が重要な課題となる。<sup>(4.19)</sup>

(c) のデータベース構築で重要な要件は、広範な利用ニーズに応えられるシステム、標準且つオープン環境を保証し拡張性を持ったシステムであることが問われる。<sup>(4.20)</sup>

### 《記》

我々が構築したデータベースのうち財務関係のデータベースは、下記のデータが収録されている（詳細は、3.2.1 「財務諸表データ」を参照）。

#### (イ) 国内企業財務データ

- ・日本興業銀行、財務データ：2,026社（1994年）、418項目。
- ・日本経済新聞社、連結決算財務データ：1,274社（1994年）、164項目。
- ・日本経済新聞社、金融財務データ：150銀行（1993年）、164項目。

(4.19) 詳細は次の資料を参照。

- ・民野庄造、「行列処理と経済経営情報分析システム」、経済経営研究叢書、経営機械化シリーズ 19、神戸大学経済経営研究所、1986、pp.115-252。
- ・本稿〈付録1〉 経済経営情報分析システム (SECRETARY) の記述文法。

(4.20) 我々が構築したデータベースは、この課題に対してデータベース記述言語群に OS 自身が持つ基本情報処理言語（コマンド・プロシージャ言語、PL/I言語等）を用いる方法を選択することで対処し大規模のデータベース（現在約 400 万系列のデータを収録）を構築することが出来た。

## (ロ) 外国企業財務データ

- S & P社 (米国) 財務データ : 米国及びカナダの約 4,800社 (1994年), 150項目。
- Extel社 (英国) 財務データ  
20カ国, 約4,800社, 英国 330項目, 英国以外 200項目。

(d) の問題に対しては, 速報性と対話的利用環境を満足する利用システムが望まれる。<sup>(4.21)</sup>

新世代の経済経営情報分析システムとして“SECRETARY”に望まれるコンセプト, 機能は次のような事項になろう。

- 利用者 (クライアント) がより能動的に行動できる提供者 (サーバ) との相互依存システム (コミュニケーション) の確立。
- ネットワーク (インターネット) 環境への対応。
- グラフィック, アナログ等ビジュアルな操作環境の強化, および, 利用機能にオブジェクト指向プログラムを取り入れ, 人の思考に接近した利用環境をより進める。

## 4.4.1c OS (メイン・フレーム : s2)

メイン・フレームのOSは情報処理機能のソフトウェアの集大成された体系

---

(4.21) 我々はこの課題に対して, データベースに関するあらゆる情報サービス, 各種利用ツールの操作法, データ提供側と利用者・利用者相互の情報交換等を行う機能を持つ“HOW (Helper of Workshop) システムを開発した。

同システムに関しては次の資料で触れた。

- 民野庄造, 「財務分析と推論言語」〈付録〉“財務データベースと分析ツール”, 経済経営研究叢書, 経営機械化シリーズ 20, 神戸大学経済経営研究所, 1989, pp.79-85。

でデータベース構築に必要とされる基本ソフトウェア、データベース管理、通信管理等の殆どの基本制御プログラムをそのOSから調達することができる。

長年の実績を持つ同OSの採用によって、信頼性・安定性が高くセキュリティを備えた大規模なデータベース・システムをつくることができる。

メイン・フレームに構築した基幹データベースを分散システムより構成されるクライアント・サーバ・システム(CSS)に組み入れるには、従来のシステムで構築した基幹DBと新しいシステムで構築した分散DBの両者のアーキテクチャを吸収するインターフェースが必要になる。

その役割を担うのが、ゲートウェイ(NOS機能を持つCSSサーバ:S1)である。

第一のインターフェースは、基幹データベースと分散データベースのデータベース管理システム相互の論理変換に関するもので、基幹データベースから分散データベースをアクセスする場合は基幹データベースの論理で見え、その逆の分散データベースからアクセスする場合は分散データベースの論理で見えるインターフェースを相互のサイトのデータベース管理システムが備えていることである。

第二のインターフェースは、メイン・フレームのOSと分散システムのOSに関するもので、ネットワーク環境の中での通信機能と端末のエミュレーション機能を行うゲートウェイを設け、分散システムあるいはユーザ(クライアント)からメイン・フレームが自分の論理で見られるようにすることである。

#### 4.4.2 サーバ側ソフトウェア階層(a-2:DB-b)

メイン・フレームのデータベースを構築・管理する“データ管理システム”は長期にわたって蓄積された資産と実績に裏付けられた信頼性、安定運用を行うことの出来るいわゆる枯れたソフトウェアであるが、それを稼働する環境は

ハードウェア、ソフトウェア共生産メーカー・特定のOS等指定のプラットフォームの下でのみ動作する。

データベース管理システム（DBMS）は、ハードウェアの生産者自身が提供するシステムとソフトウェア・ハウスが開発・提供する製品に分けられるが、現状ではデータベースシステムの構築は何れも特定のオペレーティング・システムによるクローズッド・システムとして構築されている。

クローズッド・システムで問題とされるのは、クライアント／サーバシステムにみられるような情報システム基盤の革新的な環境変化に於いても特定企業を開発母体とし、その組織内の開発理念に従ったシステム仕様の製品を使用者は今後も継続して用いていかなければならないという問題がある。

データベース・システムは、情報管理の全機能が集約された総合情報システムであるため、そのシステムを記述するデータ管理システムには“記述論理の継続性”，プラットフォームを特定しない“オープン性”が要求される。

現在情報システム開発部門は、情報システムの適用範囲の拡大、変革していく時代への対応等に追われバックログが増え続けているといわれている。

ソフトウェアの需給バランスが崩れソフトウェアが不足して社会的な問題を起す恐れを“ソフトウェア危機（software crisis）”と呼んでいるが情報化社会の進展と共に情報システムの世界を超えた問題として、ソフトウェアの生産性、流通に関連する問題として真剣に取り組まれるようになった。

このような危惧されている問題に対して、ソフトウェアのオープン化は生産性向上のための共通の開発基盤を与えるものでもあり、これからのネットワーク環境に於けるサーバ側ソフトウェアに望まれる基本理念である。

#### 4.4.2 a 分散DBサーバ（s3：DB-b）

— 財務諸表作成支援、クライアント管理、財務分析等 —

財務データベースの分散サイトとしては次のような機能を持つものが考えられる。

(a) クライアント管理

クライアントのソフトウェアのインストール、ソフトウェア環境の設定・更新等の管理機能をネットワーク環境で行う働きをサーバが持ち、クライアントへのソフトウェア配布、ソフトウェアのバージョンアップ管理等最も新しいサービスが受けられるようにする。

また、ネットワーク分散環境で稼働するクライアントの運用管理（動作状況の把握・管理・支援機能等）を行う。

(b) 財務諸表作成・管理・監査支援サーバ

経営活動によって生じた取引記録をまとめた総勘定元帳、資産、資本等を元に損益計算書、貸借対照表等の財務諸表が作成されるが、その作成・管理・監査に関する情報提供等の処理を行うサーバである。

企業グループでネットワークがつくられている場合はグループ各企業から各種の財務データを収集・編集し連結財務諸表の作成も可能である。

《記》

財務諸表を作成するサーバの場合は、仕訳帳（原データ）データベースを必要とする。

(c) 財務分析サーバ

主として財務諸表を分析の素材にし企業の財務体質を表す収益性、財務安定性（健全性）、成長性等に関して各種の数量的分析手法を用いて分析する機能をクライアントに提供する。

(d) ファイル・サーバ、プリントサーバ等

ファイル・サーバは、データベースで取り扱われないファイル、たとえば、システム開発で用いられるファイルはこのサーバで集中管理される。

プリントサーバは、ネットワーク内の各種サーバ、クライアントの多様

な出力要求を受け入れ印刷出力する。これらのサーバは、異なったOS、プロトコルを吸収するシステムが望まれる。

そのほかのネットワークサーバとしては、IPアドレス/ドメイン名を管理する“ドメインサーバ”，電子メールを支援する“メールサーバ”等が必要である。

また、財務管理関係のサーバとしては、企業活動の原資である資本に関し、資本調達・資本コスト・資本分配・資本運用等を総合的に分析・制御する“資本予算・支出等資本運用管理サーバ”，企業活動を維持し安定した企業経営を継続していくための中・長期的な見通しを立てる“投資，配当政策等意思決定支援サーバ”等の分散サイトが考えられる。

#### 4.4.2 b 分散システムのOS (s2,s1)

分散システムのOSとしては，“UNIX”，“Mach”，“Windows”等の汎用機能を持つもの、とネットワーク専用OS (NOS: Network Operating System)がある。端末エミュレータ等の制御プログラムもOSの一部としてみられる。

クライアント/サーバ・システムを基盤とするオープン環境の下で構築されるデータベース・システムは多様なリソース (ハードウェア装置，ソフトウェア製品，ネットワーク等)<sup>(4.22)</sup>が存在する。そのリソース相互は，高度のプロトコル変換機能を持つゲートウェイ，TCP/IP等のプロトコルの変換・制御を経てネットワークに複合的に結合され，リソースそれぞれの役割が集積・統合され所要の機能を持つ情報システムがつけられる。

---

(4.22) J I S情報処理用語によると、「計算機系のあらゆる機能。演算装置，主記憶装置，データチャネル，入出力装置，ファイル，制御ルーチン等。」と定義されている。

リソースは、ハードウェア側面でみると小管理単位では端末の一構成要素であるキーボードとその制御を行うデバイスドライバから、分散サイトにおいてデータを集中して管理するファイルサーバ、ソフトウェアの面でみると漢字コード変換（例えば、UNIXのEUC / PCのシフトJIS相互変換）からネットワークのプロトコル制御を行うTCP/IPまで、ハードウェア/ソフトウェア、低位/高位の機能水準等様々な管理単位に分けられる。

クライアント/サーバ・システムのリソースを役割でみると次の三つに分けられる。

#### (1) サーバ

システム運用管理用として、ファイルサーバ、ネームサーバ、メールサーバ、ネットワーク管理サーバ、ゲートウェイ、プリントサーバ等。

また、各分野のアプリケーションを支援するサーバも必要であり、財務情報分析を目的とするシステムでは財務諸表作成・管理・監査サーバ、財務分析サーバ等が必要となろう。

#### (2) クライアント

サービスを受ける側で、DOS端末、UNIX端末、Windows端末、X端末、およびその複合端末。

#### (3) ネットワーク

プロトコルに関し：TCP/IP、AppleTalk、IPX (NetWare)。

オペレーティング・システム (OS) は、それらのリソース全体の制御を統一的に行い、システムサービスの生産性と信頼性を高め、分散サイト相互間やユーザに対して高水準で一貫性のあるサービスを提供していく役割を担う。

このようにクライアント/サーバ・システム環境を制御するOSは、従来のメインフレームで用いられてきたOSとは異なった役割と機能を持つものが要求される。

第一に上げられるのは、マルチベンダ、マルチプロトコル支援等オープンな開発と実行環境を提供出来るシステムであること。従来のOSは、特定のコンピュータの下でのみ動作するシステムで且つそのOSに関する仕様が非公開のものが多かった。

伝統的情報システムの構築は、その企業または企業グループのみで行われ第三者の参入が困難な状況にありその結果として種々の問題を残すこととなった。システムを構成するソフトウェアのみに限ってみてもソフトの大半はそのOSの規格・性能に制約され細部のルーチンまで全て自前のシステムの開発が必要になること、情報の広域的利用に関して自システムとOSを異にする他のシステム間のインターフェースに個別的で特別の技術的な取り決め、運用規約等が必要であった。

システムの変更・機能拡張等を伴うとその開発・維持には予期以上のマンパワーを要しバックログを増やすことにもなる。

第二には、基幹OSはパーソナルコンピュータから大型コンピュータまでの広範囲のハードウェアのレンジを統一した論理で制御できるものが望まれる。

現在のOSは、汎用大型コンピュータではIBM MVS (Multiple Virtual Storage)系、ACOS系、中型コンピュータでは、VAX VMS (Virtual Memory operating System)、小型コンピュータではIBM OS/2, MS-DOS, Macintosh等、論理も使い方も異なっている。

これらのOSのインターフェースをとるリソースとして、端末の規格統一化を行った仮想端末プロトコル (virtual terminal protocol), 異なった各種の端末の模擬動作を行う端末エミュレータ, プロトコルの異なるネットワーク間を接続するゲートウェイ (gateway) 等がある。

以上のように機能・役割を異にし且つ分散配置されるリソースがオープン環境の下で定められている論理 (プロトコル等) に従った稼働を行うためには、システムを構成する全ての対象 (リソース) を統一した論理・手法で支援・制

御するシステム（OS）が必要である。

上のサーベイと視点からOSの具備すべき機能を整理すると、次に示すような共通の論理を提供する基盤ないしシステムが期待される。

（ア）オープンな動作環境を支援するOS（オープンシステム）

オープンシステムとは、特定のメーカー、ベンダーに依存することなく標準仕様に準拠して作られたハードウェア、ソフトウェアで構築されたシステムと定義づけられている(X/Open Company limited資料による)。

（イ）マルチユーザ、マルチタスク<sup>(4.23)</sup>、マルチスレッド<sup>(4.24)</sup>を支援するシステム。

（ウ）バッチ処理および会話方式を可能とし豊富なコマンドとコマンドプロシージャ機能を持つこと。

（エ）小型から大型コンピュータまで共通に用いることの出来るプログラミング言語（例えば、C言語）、データベース定義言語・検索記述言語（たとえばSQL）を持つこと。

（オ）国際規格、業界標準のマルチプロトコルをサポートし分散環境に対応したオープン性の高い（マルチベンダー）ネットワーク機能を持つこと。

（カ）高度のファイル管理機能とその信頼性・安全性・セキュリティーが保障されること。

（キ）高水準・高機能・操作性の高いアプリケーション・ユーザインタフェースを提供できるシステム機能を有すること。

---

(4.23) マルチタスク：プロセスと同義語、OSの制御単位を云い仮想のプロセッサをつくり見掛け上多数のプロセッサが存在するような制御を行う。

(4.24) マルチスレッド：プロセスよりリソースを細かい単位（メモリ、ファイル、タイマー等）に分割して処理を行う。一つのプロセスは複数のスレッドに分けられるためリソースの待ち時間を削減しサービス水準の向上と処理効率を上げることが出来る。

## 《注記》

これらの条件を満たすサーバ機能を持つOSとして、現有では中・大型コンピュータのシステム管理では“UNIX”，小型コンピュータに導入されるOSとしては“Windows NT”が上げられる。

UNIXは、長年の開発史(1969年 AT&Tベル研究所で開発開始)と世界規模での利用実績を持ち膨大なソフトウェアの蓄積があり、その大部分のソフトウェアがフリーウェア、シェアウェアとして誰でも入手可能である。

一面、システム(OS)のソースプログラムを公開したことにより多くの人の関心を呼びシステムの機能拡充が行われたが、結果として幾多の派生システムが生まれソフトウェアの互換性の面で問題が生じている。現在その統一仕様への取り組みが、UNIXの二大勢力を形成していたOSF(Open Software Foundation, Inc.)とUII(UNIX International), UNIXをライセンスしているX/Open Company Ltd.等で行われている。

いまセンセーショナルに新しい情報基盤として取り上げられているクライアント/サーバ・システムを現実のものにしたのはUNIXであり、関連ソフトウェアの蓄積を行ってきた多数のシステム技術者を擁しているためこれからの分散システムの基盤OSとして期待されている。

Windows NTは、UNIXが持っているCPUのマルチスレッド制御、マルチプロトコルに対応したネットワーク機能の標準組み込み、アプリケーション・プログラムの互換性、管理・運用・操作機能の容易性等先進の技術が取り入れられたサーバ、および、クライアント何れにも採用可能なOSでクライアント/サーバ・システムの新しいプラットフォームとして注目され、小型コンピュータ分野で採用されつつある。

将来的には、オープン環境を実現するため一台のコンピュータが複数のOSを持つマルチOS化が進むことになろう。たとえば基幹OSをUNIXとしX-Window(MIT開発)の一つのセッションでWindows環境を実現し、シス

テム回りの制御をUNIXで行わせ、利用者に近いアプリケーション・プログラムをWindowsの下で実行させるシステムのようなものが考えられる。

#### 4.4.3 ユーザ側ソフトウェア階層 (b)

ユーザ側（クライアント）のシステム関連のソフトウェアは、階層的に見ると次に示すようにネットワークの通信路を軸にサーバ側と対称に位置し、サーバ側とユーザ側の対応する各階層は同じプロトコルによって制御される。

- u 1 : ネットワークOS (NOS)
- u 2 : ・DOS and/or Windows  
       ・UNIX and X Window  
       ・その他のOS環境
- u 3 : SQL<sup>(4.25)</sup>/DBアクセス・マクロ等ミドル・ソフトウェア
- u 4 : アプリケーション  
       (財務分析システム, Exel, Lotus1・2・3等)

ユーザがコンピュータに情報処理を依頼するときに望むコミュニケーションの形は、人が今まで培ってきた論理・手法・手続きでコンピュータ（情報システム）と情報交換を行うことが出来ること、人とコンピュータの間で蓄積してきた関係・ノウハウ・実績がコンピュータの技術革新等の環境変化に対しても長期的に相互間の関係の連続性が保たれることである。

その意味でシステム関連のソフトウェアは、環境変化に対しても人とコンピュータのバッファの役割を果たすべきあり、人（ユーザ）の見えない位置で間

---

(4.25) SQL : Structured Query Language の略。関係データベースのデータ検索・データ更新等を行うISO（および、JIS）規格の言語。

接的に支援動作を行うシステムが望まれる。すなわち、表面ではウィンドウ環境のような感覚でコンピュータとコミュニケーションを行う仕組みを持ち、システム関連ソフトウェアは隠れた位置にありエンド・ユーザがそれに関わることなく容易に利用出来るのが望ましい形である。

今後のシステム機能として望まれるのは、ユーザ自身で行うことの出来る多様な操作環境を容易に生成・更新出来る機能を公開・提供していくことである。

ネットワークを含めたOSのシステム関連ソフトウェアとデータベース管理システム（DBMS、最近はDBエンジンとも呼ばれている）の間、および、DBMSとアプリケーションの間を関連づけるには多種多様のソフトウェア製品が要求される。そのインターフェースの役割を果たすソフトウェア（ミドル・ソフトウェア）はオープンシステムの下ではますます重要となる。国際標準・業界標準化の進展とともにそのソフトウェア市場も形成されつつある。



## 第5章 財務情報分析の計算・処理体系 と情報分析システム

企業活動、個人・社会・国と企業との関わりで生じるデータを測定し、集約・編集・加工を施して生成された財務関連のデータをデータベース等によってシステム化し、広大な財務分野の分析対象、たとえば、財務管理・投資等の意思決定に関する情報を与えること等を目的とする統合的・汎用的“財務情報分析システム”を構築するためには、システム構築に関するノウハウを持っていること、関連するソフトウェアの整備・蓄積を必要とすること、物量的な条件を満たすこと等困難な問題を伴う。

我々がこの課題に取り組むことの出来るのは情報分析システムの概念・理念、研究・開発のシステムズアプローチの手法等を提示すること、分析課題・分析対象を限定したシステムを開発しその実験成果を世に問うことは可能である。

たとえば、財務情報分析の体系化を試みることに、情報分析システムのコンセプトと構造を公開しシステム設計のノウハウを提示することも出来る。

財務諸表を中心に財務関連のデータを用いた経営分析は、19世紀の後半にその起源を見いだすことができ長い歴史をもっている。<sup>(5.1)</sup>

経営分析に関しては、財務分析、財務諸表分析、財務情報分析等のタイトルで多くの書物、研究論文が刊行されている。それらの成果は、財務理論を規範にしその分野の研究者・専門家によってまとめられたものである。

コンピュータを用いた財務情報分析システムにおいて、分析システムで何が出来、また分析能率が上げられるかは分析システム自身が持つ言語機能（機械との対話環境を含めて）に依存する。

---

(5.1) 後藤幸夫・山上達人編、「テキストブック経営分析」、有斐閣、1979年。

一方、現在のノイマン型コンピュータの特性は、逐次（連続）処理、計数把握、決定論的、プログラム（手続き）的であるのに対し、人間思考の特徴は、突発的（ランダム）、連想・パターン把握、非決定論的（経験・確率）、学習的でありノイマン型コンピュータの論理構造と人間思考との間は根本的な論理ギャップがある。従って、コンピュータを用いて問題を情報システム化する段階では現実の問題をコンピュータ向きの論理に翻訳・整理する過程が非常に重要なファクターとなる。

上に示したように財務情報分析の問題に対しては、財務論・財務分析理論、コンピュータ・サイエンス、情報システム論、人間/機械インターフェース等多面的な関わりとして財務情報分析を捉えていく必要性を筆者は痛感している。この認識から財務情報分析の問題をアルゴリズムの視点で捉えその類型化を試み、実証面（主として SECRETARY を用いて）をふくめ若干の考察を行うことにしたい。

ここでの数理の定義、および展開は情報分析システムの規範に従って記述することにする。

## 5. 1 一般的データ操作機能（データ通覧・検索・編集加工・表出力等）

財務情報分析の素材はデータであることから分析で必要とするデータを探索・見極めそのデータを取得する作業は分析過程の中でも重要な位置を占めそのために費やす時間は分析全体からみても高い比率となっている。

したがって、分析の省力化、能率を高めるには情報分析システムの機能としてデータ探索・データ検索を速やかに行うことが出来、データ操作機能としても分析で用いるデータの特性にマッチしたデータ操作、演算、表等の出力機能を備えたものが要求される。

財務諸表を対象とする情報分析システムでは、次に上げるようなデータに関

する基本的な操作機能を備えていることが期待される。

- 複数のデータベース（国・分野別等編集機関の異なるデータ）から取り出された所要データ全てをランダムに扱い編集出来る記憶空間を備えること（国際比較分析，デフレータによるデータ修正・補正を行う時等に必要である）。
- データ検索機能として各種条件指定による検索，および一組の企業・一組の財務項目の情報を与えることにより複数企業・複数項目のデータが一命令で検索できる機能。
- 演算機能としては，スカラー，ベクトルのみならず二次元（配列等）の演算命令を持ち財務分析のためのより多くの組込み関数を持っていること。
- データ処理の繰り返し制御，および一括処理（バッチ処理）機能。
- 財務諸表分析では特定時点の諸財務項目を用いたクロスセクション分析と時系列分析を可能とする代数演算機能，および表出力機能を持つこと。

[一般的データ操作機能の実行例]

(a) 財務諸表出力（その1）データレコードの検索と表出力

（ANAL01：付録2参照）

(b) 財務諸表出力（その2）貸借対照表：資産の部・負債の部

（ANAL02：付録2参照）

(c) 財務諸表出力（その3）損益計算書

（ANAL03：付録2参照）

## 5. 2 比率分析・百分比比率分析

比率分析 (ratio analysis)・百分比比率分析 (100 per cent-) は、伝統的な経営分析のための代表的手法で財務分析では主として財務諸表の各項目を用い

企業の財務健全性（財務流動性・財務安全性）、成長性等の分析を行うときの指標算出に用いられる。

比率分析は経営比率の外、ある一時点を基準にして算定される“指数”を用いてデータの趨勢分析を行う方法。

項目グループ内のある基準項目（合計値等）を100として他の項目の百分率をとり基準項目に対する割合（構成比）を算定し、規模の異なる企業相互の比較分析等に用いられる。

### （1）比率分析の数理表現

スカラー値の場合の比率の定義式は次式で表される。

$$\frac{b}{a} \quad \text{百分比の場合は, } \frac{b}{a} \times 100 (\%) \quad (\text{m 5.1})$$

データが表（行列）で表される場合の定義式は：

$$\frac{\mathbf{b}}{\mathbf{a}} \quad \text{百分比の場合は, } \frac{\mathbf{b}}{\mathbf{a}} \times 100 (\%) \quad (\text{m 5.2})$$

ここで  $\mathbf{a}$ 、 $\mathbf{b}$  は行列、ベクトル又はスカラーであり、行および列を示す指標は別に指定される。

$\mathbf{a}$  が行ベクトルの場合、分子  $\mathbf{b}$  行列の列ベクトルの各要素と  $\mathbf{a}$  ベクトルの対応する要素との比率がとられる、構成比はこのケースになる。

情報分析システムでは、 $\mathbf{a}$  が行列かベクトルかは情報分析システムが持つ変数辞書により  $\mathbf{a}$ 、 $\mathbf{b}$  変数の属性で識別される。

前年（或いは前期）を基準とする増加率（成長率）は：

$$\left[ \sqrt{\frac{a}{a_{-1}}} - 1 \right] \times 100 (\%) \quad (\text{m 5.3})$$

n年間平均の増加率（成長率）は：

$$\left[ \sqrt[n]{\frac{a}{a_{-n}}} - 1 \right] \times 100 (\%) \quad (\text{m 5.4})$$

## (2) 適用対象の財務分析領域（分野）

財務分析（financial analysis）は、主として財務諸表を用い下記に示すような企業等の収益性、財務健全性、成長性等の経営分析を行うことをいう。

- ・収益性分析：財務諸表を用いた資本利益率等収益力の度合いに関する分析。収益性は、資本・収益・費用等のデータを用い、使用総資本利益率（純利益／期首・期末平均総資本）・自己資本利益率（純利益／期首・期末平均自己資本）等の利益率を算出し収益性を判断する指標として用いられる。
- ・企業の財務健全性分析：財務流動性・財務安全性・資金運用分析等、企業の支払い能力等財務の健全性の程度に関する分析。

財務流動性は、流動負債に対する流動資産の割合で示され短期的（1年以内）な支払能力の性質を、財務安全性は負債の自己資本に対する比率で計られ長期的支払能力に関する分析である。

- ・そのほか、売上高・総資産等の増加率で求められる成長性分析、付加価値額（企業活動によって新たに生み出された価値額に減価償却費を足した額）の伸び率、付加価値生産性（付加価値額／従業員数）等の指標を用いた付加価値分析等が上げられる。

## (3) 応用例（付録2“財務分析問題（第5章）実行例”を参照）

### (a) 収益性分析（ANAL04：付録2参照）

資本回転率と売上高利益率の計算と結果の出力を行う。

```
3 >> GETR ¥REN05IA REPEAT 6502:
```

連結決算データベース（NEFA）より指定企業（証券コード：6502）の財務各項目（カタログ名：¥REN05IAのコード・リスト）の検索。

```
4 >> LET @NOI=NOWD(¥REN05IA):
```

“NOWD”は、カッコ内に指定するカタログの内容の語数をカウン

トする組込関数，即ち，指定検索項目数が求まる。

8 >> LET RATIO1 = #(@SALES) / #(@CAPITAL);

資本回転率を求め“RATIO1”に入れる。

“#”記号はアクティブ・エリアを示す。@SALESおよび，@CAPITALはアクティブ・エリアの位置を指し示すスカラー変数。

(b) 流動性・安全性分析 (ANAL05: 付録2 参照)

流動比率と自己資本比率の計算と結果の出力を行う。

(c) 成長性分析 (ANAL06: 付録2 参照)

売上高増加率，経常利益増加率（3年平均）の計算と結果の出力を行う。

8 >> LET RATIO1 = (#(@S) / #(@S,,-1) - 1) \* 100;

9 >> RATIO2 = (#(@G) / #(@G,,-3)^(1/3) - 1) \* 100;

“#(@G,,-3)”の-3は，時系列データの3期ラグを示している。

!記号はべき乗を表し，“!(1/3)”は立方根を算出する。

### 5. 3 利子率関数に関連する分析

貨幣価値は利子率（引用のケースによって収益率・割引率・資本化率とも呼ばれる）の関数で表される。財務関連の数量分析において利子率は重要なキーワードの一つであり終価・現在価値・内部収益率・年金現価等の計算式の中で定数，または変数として用いられ予算・資金運用・投資・資本コスト・企業評価等財務の広い分野の分析アルゴリズムに組み入れられる。

企業の財務分析といえば総合且つ系統的な分析となるが，PCで手軽に使うことの出来る表計算システムでは，下記の数理表現，あるいはその変形応用として期間・キャッシュフロー・利子率に関して，与える情報と取得する情報の

選択技を幅広くした財務関数を多数用意し突発的・個別的な問題に対しても容易に使うことが出来るシステムがつくられている。

モデル(式)は、貨幣(資金)・利率・期間を表す演算項で構成され、そのうち期間は通常単一の数値(スカラー)で表されるが、資金と利率に関しては初期投資等のスカラー値を除きキャッシュ・フローのように時間系列の値としてフロー(ベクトル)で表されるのが一般的である。

また、不確実性環境下で資金・利率のデータを捉え、将来を予測するというような場合は、キャッシュフローを確率データとして扱う方法が用いられる。

#### (1) 数理表現<sup>(5.2)(5.3)</sup>

##### (a) 終価計算(利息計算), 利回り計算

元金を $P$ , 複利終価を $S$ , 利率を $r$ とすると,  $n$ 年後の $S$ は次式で表される。

$$S = P(1 + r)^n \quad (\text{m } 5.5)$$

また, 利回り計算の場合は,  $S$ ,  $P$ ,  $n$ が既知で $r$ を求めることであり

$$\frac{S}{P} = (1 + r)^n \quad \text{より,}$$

$$r = \sqrt[n]{\frac{S}{P}} - 1 \quad (\text{m } 5.6)$$

##### (b) 現在価値法

現在価値法(PV: Present Value method)は、投資の経済性判定、資金運用等に用いられ将来の貨幣価値を現在値に換算する方法である。貨幣の流入・流出のバランスで見える場合を正味現在価値法(NPV: Net ---)と

(5.2), (5.3) 参考文献

(1) 諸井勝之助著, 「経営財務講義」, 東京大学出版会, 1979。

(2) 生駒道弘・小野二郎・濱本 泰編, 「テキストブック経営財務」, 有斐閣, 1985。

呼ぶ。

いま、投資からもたらされるインフローの現在価値を  $PVF$  とし、また投資のために支出した投下資本のアウトフローの現在価値を  $PVI$  とすると正味現在価値は次式で表される。

$$NPB = PVF - PVI = \sum_{t=0}^n \frac{F_t}{(1+r)^t} - \sum_{t=0}^n \frac{I^t}{(1+r)^t} \quad (\text{m } 5.7)$$

アウトフロー（投資）が最初の一回だけだとすると  $NPV$  は次式で表される。

$$NPV = \sum_{t=0}^n \frac{F^t}{(1+r)^t} - I_0 \quad (\text{m } 5.8)$$

現在価値法による投資案の採択の基準は、 $NPV$  がプラスの場合採択されマイナスの場合は却下される。

また、インフロー（ $F$ ）が各期一定であるとすれば、 $NPV$  は次式で示される。

$$NPV = F \cdot A - I_0 \quad (\text{m } 5.9)$$

ここで  $A$  は、年金現価係数と呼ばれ次式で示される。

$$A = \frac{(1+r)^n - 1}{r(1+r)^n} \quad (\text{m } 5.10)$$

年金現価係数は、期末に一定額のと例えば年金の掛金を複数年にわたって支払うものとし、その総額を現在時に一括して支払った場合の額を求める時にこの係数が使われる。

また、ある額の投資（初期投資のみ）を行い、毎期末に定額を回収するとし  $n$  期末後の複利終価額全てを回収する問題の例を考える。投資額の現在価値合計を  $V$  とすると毎期末の回収額  $C$  は次式で表される。

$$C = V \cdot B$$

ここで  $B$  は、資本回収係数と呼ばれ次式で表される。

$$B = \frac{r(1+r)^n}{(1+r)^n - 1} \quad (\text{m } 5.11)$$

年金現価係数と資本回収係数は、式 (m5.10), (m5.11) に示すように逆数の関係になっている。

また、キャッシュ・フローの値を確定したデータではなく不確実なデータとして扱い確率論を導入したキャッシュ・フロー分析もよく知られ投資決定等に応用されている。

キャッシュ・フローが確率変数のときは正味現在価値も確率変数となりその期待値からの誤差は正規分布をすることが知られている。

したがって、正味現在価値の標準偏差が求められれば、ある任意のキャッシュ・フローの正味現在価値の誤差確率は正規分布から近似させることが出来る。<sup>(5.4)</sup>

実際は、次の式により確率変数の正規化 (平均=0, 標準偏差=1) を行うことにより正味現在価値の任意の値 (X) に於ける実現確率 (累積確率) を、標準正規分布表、または分布関数 (誤差関数) により求めることが出来る。

$$Z = \frac{X - E(NPV)}{\sigma(NPV)} \quad (\text{m } 5.12)$$

ここで、“X” は任意の正味現在価値、“E (NPV)” はNPVの期待値、“σ (NPV)” はNPVの標準偏差。

#### (c) 内部収益率法 (IRR: Internal Rate of Return)

投資案のキャッシュ・アウトフローの現在価値の合計とキャッシュ・インフローの現在価値の合計を等しくする割引率を内部収益率と呼び、利子率とは独

(5.4) キャッシュ・フロー分析に関しては、次の拙稿で推論言語“PROLOG”を用いて記述し若干の考察を行った。

民野庄造, 「財務分析と推論言語」, 経済経営研究叢書, 経営機械化シリーズ 20, 神戸大学経済経営研究所, 1989, pp. 49-74.

立して決定される。

数式は次式で表わされる。

$$\sum_{t=0}^n \frac{I_t}{(1+r)^t} = \sum_{t=0}^n \frac{F_t}{(1+r)^t} \quad (\text{m } 5.13)$$

資本支出が現時点で1回だけなされ、資金流列が1期末以降に流入する場合は次式で表される。

$$NPV = \sum_{t=1}^n \frac{F_t}{(1+r)^t} - I_0 = 0 \quad (\text{m } 5.14)$$

## (2) 適用対象の財務分析領域 (分野)

財務分析は経営活動の中で計測・整理・編集され報告される各種の財務諸表を分析の素材とし、企業活動の意思決定に役立つ情報を提供することが目的であり、研究・開発で積み重ねられてきた各種の分析手法を分析者が選択・駆使し評価と決定行動の情報を獲得することである。素材とされるデータは分析領域・過程の全てにわたって経済的に共通の尺度で計測・伝達され、処理し得る単位でなければならないということで通常貨幣が選択される。

貨幣価値は、利率と時間の関数で表され経営活動の諸々のイベントで生じるデータに対して(1)で示した算定式類によりその計測・予測が行われる。

ここで示したアルゴリズムが適用されるのは経営活動を趨勢で捉える場合であり経営の中・長期的な計画ないし経営(財務)分析の広範囲の問題領域で用いられる。

資本予算、投資決定、配当政策、企業評価等の理論はこのアルゴリズムをベースにして数理的に展開される。

(3) 応用例 (付録2 “財務分析問題 (第5章) 実行例” を参照)

(a) キャッシュ・フロー分析 (その1) (ANAL07: 付録2 参照)

—資本支出最初時のみ, 正味現在価値 (NPV) 算定—

5 >> PERIOD @STY TO @EDY:

分析期間の定義をスカラー変数 (@STY, @EDY) で指定。

7 >> LET @RR=@RR \* REN07R1(@YY);

各期の総合利子率 (利子因子) の計算。

11 >> LET REN07Q=CUMU(REN07A):

各期末の現在価値を求める。“CUMU” は累積値を求める関数。

(b) キャッシュ・フロー分析 (その2) (ANAL08: 付録2 参照)

—資本支出最初時のみ, NPVがプラスになる期を求める—

4 >> @NPV=-@INVEST:

正味現在価値を入れる変数 (@NPV) に初期投資額 1000 をマイナスで設定。

9 >> DO @YY=@STY TO @EDY WHILE(@NPV<0):

正味現在価値がマイナスの間中 (WHILE(@NPV<0)が真) 文を繰り返す。

条件判定が“偽”となったときの“@YY値-1”が求める期で, 本例題では2期目である。

17 >> DTM REN07CFA, REN07R1, REN07RR, REN07A, REN07P:

関係データの表示。

それぞれ, キャッシュフロー (REN07CFA), 各期の利子率 (REN07R1), 各期の総合利子率 (REN07RR), 期内の現在価値 (REN07A), 各期の正味現在価値 (REN07P) を表している。

## (c) キャッシュ・フロー分析 (その3) (ANAL09: 付録2 参照)

——キャッシュ・フロー (CF) の正規分布による分析——

この例題では、各期5ケース、3期のキャッシュ・フロー・データを入力データとして用い、確率論を導入し投資プロジェクトの正味現在価値がマイナスとなる確率を求める。

但し、各ケースの確率は各期共同じとし、各期のキャッシュ・フローは互いに独立であると仮定して分析を進める。

9 >> GB REN07FA TO REN07FC:

ベース・エリア (BASE) に登録済みの3期分のキャッシュ・フロー・データ (REN07FA, REN07FB, REN07FC) をアクティブ・エリアに呼び出す。

GB は、GETBASE の省略形。

10~13 >> DO @K=1 TO 3:

LET @AV=MEAN(#(@K));

#(@COUNT+1)=(#(@K)-@AV)\*(#(@K)-@AV): END:

各期 (3期分) のキャッシュ・フローの分散を求める。“MEAN” は、括弧内で指定するデータ系列の平均値を求める関数。

“@COUNT” は、現時点のアクティブ・エリアのレコード数が入れられるシステム変数。

“#(@COUNT+1)” は、アクティブ・エリアの空いている領域で @COUNT を越える作業空間は動作域として自由に使うことが出来る。

14 >> LET W1=POWM(@R):

“POWM” は、括弧内のデータ系列の累乗を算出する関数。この例では、将来の利子因子 (@R) の総合値を求めている。

16 >> LET W2(@K)=EXPECT(#(@K),REN07PW);

キャッシュ・フローの各期 (3期分) の期待値を求め系列名 “W2” に

ベクトルとして入れる。“EXPECT”は、期待値を求める関数で被データ系列と確率変数の系列を入力の入数として与える。

例では、“#(@K)”が被データ、REN07PW”が確率の入れられているベース・エリアの名前。

17 >> @NPVE=@NPVE+W2(@K)/W1(@K);

正味現在価値の期待値を求める。

18 >> LET @NPVS=@NPVS+EXPECT(#(@K+3),REN07PW) / @R!(@k\*2):

正味現在価値の分散 (@NPVS) を算出する (標準偏差は “20>>”)。

この値を求めるには、“12>>”で計算した各期のキャッシュ・フローの確率分布の分散値を用いる。

23 >> LET @Z=(0-@NPVE)/@NPVS;

キャッシュ・フローの正味現在価値の期待値 (@NPVE), および正味現在価値の標準偏差 (@NPVS) で正規化 (平均=0, 標準偏差=1) を行い値 “z (任意の正味現在価値の期待値からの偏差÷正味現在価値の標準偏差)” を算出する。

この例題では、正味現在価値がマイナスとなる確率を求めるため任意の正味現在価値を “0” で与えている。

24 >> LET @P=0.5+TBLN(@Z):

“@P” が正味現在価値がマイナスとなる確率。

“TBLN” は、z の値から標準正規分布の確率を求める関数。

求めた確率が標準正規分布の中心 (ゼロ) より左の部分 (マイナス) であるため0.5で調整を行っている。

#### 5. 4 損益分岐点分析とその同族 (資本回収点分析)

分析理論のアルゴリズムは通常数式によって表されるが、数理や形式言語に

よって表現・展開される抽象概念、たとえば数理展開の経過の軌跡情報、現象・結果の相互の関係情報等を具体的な形に変換（具象）し視覚で捉えられる形（表・グラフ）で表すことも財務分析ではよく用いられる。とくにグラフ表示は情報が感覚で捉えられるので論理をつくり理解するための前提条件（知識）を必要としない、言語圏による障壁がなく理解基盤が共通である、情報密度が高い等の特徴を持つ情報の表記媒体である。

経営財務分野で用いられる“損益分岐点分析”，“資本回収点分析”等は、グラフ表示が理論の全容を明確に示しており数式による記述とそのグラフ表示が一体となり理論を説明している。

## （１）数理表現

### （a）損益分岐点分析の例

損益分岐点分析は、一定の費用構造のもとで操業度（売上高）と利益との関係を数式、または図表で表わす分析手法で通常大綱的利益計画において用いられる。

表される図表を損益分岐点図表（break-even chart）、あるいは、費用（Cost）、売上（Volume）、利益（Profit）に関する分析（CVP関係）のためCVP図表とも呼ばれている。この図表は、縦軸に金額（売上高、費用、損益等）をとり、横軸に操業度（通常売上高）をとった直交座標上に売上高と費用の一次線を同図表に描いたものである。

いま、固定費（fixed cost）を  $f$ 、変動費（variable cost）を  $V$ 、 $f$ 、および  $V$  の発生期間の売上高を  $s$ 、利益を  $g$  とすると次の関係が成り立つ。

$$g = s - (V + f) = s(1 - V/s) - f$$

式中の  $V/s$  は、変動費率と呼ばれている係数（売上高1単位当たりの変動費）でこれを  $v$  とし、売上高が変わっても変動費率は変わらないものとするれば、

$$g = s(1 - v) - f \quad (\text{m } 5.15)$$

したがって、 $s$  は、

$$s = (g + f) / (1 - v) \quad (\text{m } 5.17)$$

損益分岐点（売上高）は利益（ $g$ ）が0の時の売上高であるから次の式で表される。

$$s_0 = f / (1 - v) \quad (\text{m } 5.18)$$

また、損益分岐点分析では損益が0となる売上高をどの程度上回っているかを示す次の安全余裕率が企業の安全性の一つの指標として用いられる。

$$\text{安全余裕率} = (\text{当期売上高} - \text{損益分岐点売上高}) / \text{当期売上高} \quad (\text{m } 5.19)$$

#### (b) 資本回収点分析の例

資本回収点は、総資本が一回転するときの売上高のことで、売上高が資本の額と等しくなる点をいい、それに関連した分析を資本回収点分析という。

CVP図表と同じように、資本と売上高の関係を線形図で表したものを資本図表（capital graph）と呼んでいる。

損益分岐点分析では、費用を固定的費用と変動的費用に分け収益との関係を見るのに対し、資本回収点分析は資本を固定的資本と変動的資本に分け売上高との関係を分析する。

また、CVP図表と資本図表を併用した図を描くことにより、収益・費用・資本・売り上げの関係を分析することが出来る。

“利益図表（CVP図表）は、一定期間の費用・収益・利益・に関するフローの関係を示しているのに対し、資本図表は必要な資本ストックと操業度との関係示している点が異なる。”<sup>(5.5)</sup>

(5.5) 長浜穆良, 「財務計画と予算統制」, 生駒道弘. 小野二郎・濱本 泰編, 「テキストブック財務」第2章, p.28, 有斐閣, 1985.

## (2) 適用対象の財務分析領域 (分野)

損益分岐点分析は、一定の費用構造のもとで操業度(売上高)と利益との関係を図表で表わし、売上高、費用、利益相互の関係、たとえば、目標利益達成のためには操業度を如何なる水準に保てばよいかとか、売上げ・費用の構成が変わった場合に利益がどう変わるか等を数式や図表を用いて分析する。

損益分岐点分析は多くの前提の下に成り立っており通常大綱的利益計画において用いられる。

損益分岐点分析(CVP図表)は売上高・費用の変化と利益との関係を表すが、資本に関しては何も触れられていない。売上高(操業度)の変化と資本との関係について分析するのが資本回収点分析(資本図表)である。

CVP図表と資本図表を同一平面に表すことによって、ある目標資本利益率(利益/資本)を達成するための売上高を求めること、また、その時の費用や資本の水準を図表や数式から求めることが出来る。

## (3) 損益分岐点分析, 資本回収点分析の例

### (a) 損益分岐点分析(その1) : データ検索・出力, 分析前処理

(ANAL10 : 付録2参照)

売上高・費用関係財務項目をデータベースより検索し、続いて、変動費、および固定費を算出し損益分岐点売上高、安全余裕率を求める。

1 >> CFILE KN1A:

指定のデータベース(KN1A:興銀財務データ)をカレント・ファイル(ファイル指定を省略出来るファイル)とする。

3 >> GETR ¥¥REN06IA REPEAT ¥¥REN06KA:

データベースより指定レコードを検索する。当該コマンドは、財務項目と企業を別々に指定して検索するもので、項目・企業はそれぞれコード集合で与えることが出来る。名前(名標)の頭に“¥¥”記号の付いたもの

はカタログ名と見なされ、項目コード集合・企業コード集合の記録された情報がカタログより展開されコマンドに引き渡される。

#### 4 >> OPEN TMPF:

一時的ファイル (TMPF) のオープンを行う。

TMPFファイルには、GET系コマンドで検索されたレコードが常に入れられる。

#### 5 >> READTMPF:

GETR文(3>>)で検索した最初の企業(企業コード:9201)の全項目をTMPFファイルからアクティブ・エリア(ACTIVE)に読み込む。

#### 6 >> DO WHILE(FLAG(1)=0):

文のカッコ内の条件(FLAG(1)=1は、TMPFファイルの終わり検出時)が“真”の間中“END”文迄の文を繰り返す制御文。

#### 7 >> READSUM:

2番目以降の企業に付いて、その企業の全項目をTMPFファイルから読みアクティブ・エリアに累積する。

DO文(6>>)の制御によって各項目の累積値が求められる。

(b) 損益分岐点分析(その2): 企業群の累積値データ出力

(ANAL11: 付録2参照)

(c) 損益分岐点分析(その3):

(1) 変動費・固定費算出, (2) 損益分岐点売上高・安全余裕率算出

(ANAL12: 付録2参照)

#### 4~9 >> DO @I=1 TO 12: .....

変動費(変数名: COSTV), および固定費(変数名: COSTF)の算出。

#### 10 >> BEP=COSTF / (1 - COSTV / #(15));

損益分岐点売上高の計算。

#### 11 >> MSR=(#(15)-BEP) / #(15):

安全余裕率の計算。

### 5. 5 財務分析における多変量解析手法<sup>(5.6)</sup>

上に述べた比率分析・百分比比率分析，利子率関数に関連する分析等で用いられるデータは財務諸表等のたかだか二つの項目，あるいは変量でそのデータ項目相互の関係を調べたり項目の時間的な動きを捉えて財務の収益性や成長性について分析する手法であった。財務諸表は経営活動の結果生じた情報を総合且つ体系的にまとめた表であるように，一つの財務項目をとっても多数の関連項目・指標から算定される。したがって，実際の経営事象を分析し把握するには多くの財務データを対象にデータ相互の関係にも注目し事象との関係を解明していくことが重要である。

このような背景から，財務を総合的・体系的に捉えるための一つの方法として多変量解析が採り入れられている。

財務分析に用いられる多変量解析の手法としては，項目・比率等多数の変量の中から総合特性値に要約を行う“主成分分析（principal component analysis）”，二つ以上の母集団から抽出される多変量データについて分析し被試行体が何れの集団に属しているかを判別関数を求めて分析（倒産企業の分析等に用いられる）を行う“判別分析（discriminant analysis）”，対象間の類似性あるいは非類似性の測度を手がかりにその背後にある構造を直観的にわかりや

---

(5.6) 本項の分析問題記述例は，次の拙稿を加筆・修正したものである。

民野庄造，「行列処理と経済経営情報分析システム」，経済経営研究叢書，経営機械化シリーズ 19，神戸大学経済経営研究所，1986，pp.115-252。

また，数理の展開に関しては下記の文献を参考にした。

芳賀敏郎／橋本茂司著，「回帰分析と主成分分析」，6 主成分分析，日科技連，1980，pp.161-172。

すい形で表現する手法<sup>(5.7)</sup>で“多次元尺度法 (MDS: Multi-Dimensional Scaling)”, 分類基準が予め与えられていない場合に多次元空間内のデータを解析して類似性のあるデータをまとめていく“クラスター分析 (cluster analysis) 等が上げられる。

### (1) 主成分分析の数理表現

互いに相関のある  $p$  個の変量から  $m < p$  個の相関の無い総合化された特性値 (主成分) に要約し, その変量を用いて分析する手法を主成分分析と呼ぶ。

主成分分析は, たとえば因果関係の強い経済体制下で類似変動を示す経済データの要約 (経済指標の作成), アンケート, テスト, あるいは自然現象の観測等で得られたデータの分析にその手法が用いられる。

その分析の論理の説明と, 分析言語 (SECRETARY) による手順を説明する。

まず, 観測データは変量の尺度を統一することと分析の展開を単純化するために下式を満足する標準化データ行列 (平均が 0 で, 標準偏差 1 の系列) に変換する。

$$\sum_{k=1}^n X_{ik} = 0 \quad (m \ 5.20)$$

$$\frac{1}{n} \sum_{k=1}^n X_{ik}^2 = 1 \quad i=1, 2, \dots, p \quad (m \ 5.21)$$

標準化データによれば,  $X_j$  と  $X_k$  の相関係数は,

$$r_{ij} = \frac{1}{n} \sum_{k=1}^n X_{ik} \cdot X_{jk} \quad (m \ 5.22)$$

と簡単に表される。

---

(5.7) 林知己夫編, 柳井晴夫・高根芳雄著, 「多変量解析法」, 7章 多次元尺度法, 朝倉書店, 1985, p.141。

ここで、 $P$ 個の変量から次の1次式を仮定する。

$$Z_k = a_{1k}X_{1k} + a_{2k}X_{2k} + \cdots + a_{pk}X_{pk} \quad k = 1, 2, \dots, n \quad (\text{m } 5.23)$$

主成分分析では、回帰分析のように外生変数を持たないため、 $X_1, \dots, X_P$ の特性値を求めるための基準として、 $Z$ と $X_i$ との相関係数( $R_i$ )を採り入れ、次式に示す $\sum R_i^2$ を最大にするように $Z$ のパラメータを決定しその合成変量を説明力の基準にする。

$$R_i^2 = \frac{\left( \frac{1}{n} \sum_{k=1}^n X_{ik} \cdot Z_k \right)^2}{\frac{1}{n} \sum_{k=1}^n Z_k^2} \quad (\text{m } 5.24)$$

最初に求めた $Z_1$ は、元の変量の総合特性を表しており、その値を第1主成分と呼んでいる。

同様の方法で、第2主成分( $Z_2$ )が求められる。 $Z_2$ を算出するときは $Z_1$ との相関が0(無相関)という条件の下で $r_2(X_2 Z_2)$ が最大になるように係数を決定する。

実際には、 $Z$ の係数( $a_{ik}$ )を求めることは、行列の固有値問題に帰し次の方程式を満足するスカラー“ $\lambda$ ”と0でないベクトル“ $a$ ”を求める問題になる。

$$Ra = \lambda a = (\lambda I) a \quad (\text{m } 5.25)$$

$$(R - \lambda I) a = 0 \quad (\text{m } 5.26)$$

0でないベクトル“ $a$ ”を求めるには次の式が成立しなければならない。

$$|R - \lambda I| = 0 \quad (\text{m } 5.27)$$

この式は固有方程式と呼ばれ、 $\lambda$ の $n$ 次多項式となるから $\lambda$ は全部で $n$ 個求められる。

$\lambda$ に対応した行列 $R$ の固有ベクトル“ $a$ ”は、それぞれの $\lambda$ に関して式

(m5.26) の連立方程式を解くことにより  $\lambda$  に対応するベクトルを求めることが出来る。

### (2. 1) 主成分分析例題 (ANAL13: 財務比率の総合1: 付録2参照)

財務諸表データを用いた比率分析は手計算でも簡単に行うことが出来、古くから企業の収益性、財務健全性、成長性等の分析のために用いられてきた。

比率分析では分母子の財務項目の時間的整合性をとるとか、個々の項目の厳密な定義を行う等のために税や減価償却費等で項目を調整し比率を算定することも行うが、伝統的な比率分析では基本的に単変量による分析である。

また、貸借対照表・損益計算書等の財務諸表は経営活動の成果を集約し均衡的に表した出力表で項目相互の相関の高いものも多く含まれている。

類似項目の要約を行い総合的視点で分析を行ってみようというのが多変量解析の特徴である。実際に行われている分析では数十種の比率・成長率等の係数を入力し数種のお互いに相関を持たない主成分に要約して評価が行われる。ここでは、5 指数を入力情報とし主成分分析の方法論を提示し、併せて若干の財務分析結果の評価を行ってみることにする。

用いた 5 指数は次の比率である。

$$(1) \text{ 総資本回転率} = \text{売上高} / \text{資産合計}$$

$$(2) \text{ 売上高営業利益率} = \text{営業損益} / \text{売上高}$$

$$(3) \text{ 流動比率} = \text{流動資産} / \text{流動負債}$$

$$(4) \text{ 当座比率} = \text{当座資産} / \text{流動負債}$$

$$(5) \text{ 総資本経常利益率} = \text{経常損益} / \text{資産合計}$$

・項目とアクティブ・エリア (# 表示) の対応関係

#(1): 流動資産, #(2): 当座資産, #(11): 資産合計, #(13): 流動資産, #(17): 売上高, #(23): 営業損益, #(24): 経常損益。

・ #(31) - #(35) に、総資本回転率～総資本経常利益率が入られる。

## 《記》

- 説明は、出力結果のコマンド表示（行番号>>）と対応づけて行う。
- よく使うコマンドは省略形を用いる。
- 前コマンドをセミコロン（;）で終わると、次のコマンド名を省略することが出来る。
- 名前（名標）の後にアポストロフ（'）を付けた行列は転置行列、同様に、ダブルアポストロフ（"）を付けた行列は逆行列を表す。
- 印刷出力をページ内に収めるため行列の一部分のみ出力する場合がある。

## 1&gt;&gt; CFILE KN1A:

指定のデータベース（KN1A：興銀財務データ）をカレント・ファイル（ファイル指定を省略出来るファイル）とする。

## 2&gt;&gt; GETR ¥¥KGIT01 REPEAT ¥¥KGCO01:

データベースより指定レコードを検索するコマンド。このコマンドは、財務項目と企業を別々に指定して検索するもので、項目・企業はそれぞれコード集合で与えることが出来る。名前（名標）の頭に“¥¥”記号の付いたものはカタログ名と見なされ、項目コード集合・企業コード集合の記録された情報がカタログより展開される。

## 3&gt;&gt; PERIOD 84 TO 85:

データベースに収録されたデータの利用期間を設定する。この例題では特定時点（1984年）のクロスセクション・データが用いられる。

## 5&gt;&gt; OPEN TMPF:

一時的ファイル（TMPF）のオープンを行う（TMPFファイルにはGET系コマンドで検索されたレコードが常に入れられる）。

## 6&gt;&gt; DO WHILE(FLAG(1)=0):

文のカッコ内の条件（FLAG(1)=1は、TMPFファイルの終わり検

出時) が“真”の間中“END”文迄の文を繰り返す制御文。

7>> READTMPF:

一企業の全項目 (GETR文のREPEATキーワード迄に書かれたコード群) をTMPFファイルからアクティブ・エリア (ACTIVE) に読み込む。

8~16>> LET # (31) = # (17) / # (11); ..... # (35) = # (24) / # (11):比率計算。

#は、アクティブ・エリアのレコード指定記号。# (31) は、アクティブ・エリアの31番目のレコードを指す [# (36)~# (42) は、引用外データ・レコード]。

17>> IF @K=1 THEN MAT MMX = # (31,35/1):

条件 (@K=1) 付き“MATRIX (MATは省略形)”文。

DO文の第一回目のループ (@K=1) で、アクティブ・エリアの31-35行、1列の列ベクトルを“MMX”の名前で登録する。

18>> MAT MMX = # (31,35/1):

IF文で条件が“真”の時 (実行時) 分析システムがその文を表示する。

19>> IF @K>1 THEN MAT MM2 = # (31,35/1):

DO文の二回目のループ以後 (@K>1), アクティブ・エリアの31-35行、1列の列ベクトルを“MM2”の名前で登録する。

20>> IF @K>1 THEN MAT MMX = SUBMX(MMX//MM2):

DO文の二回目のループ以後 (@K>1), “SUBMX”関数 (小行列関数) を用いて行列“MMX”と行列 (列ベクトル) “MM2”を列で結合 (“//”記号。行で結合する場合は“%”記号) し、行列“MMX”に入れる。

文7からこの行列結合文までを企業数回繰り返すことにより全企業の財務比率 (5係数) のマトリックスが“MMX”に入れられる。

2 3 ~ 2 4 >>

DO文範囲内のIF文で、条件が“真”の時に実行された文のリスト。

2 5 >> DMX SUBMX(MMX/1,-5/1,-5/):

“SUBMX”関数を用いて行列“MMX”の1-5行、1-5列のデータを印刷出力する。

2 6 >> IU MMX,MM2:

行列“MMX”および“MM2”の登録情報を出力するコマンド（IUは、INFOUSERの省略形）。

2 7 ~ 2 8 >> I ¥¥KGIT01; ¥¥KGCO01:

検索する財務項目のコード集合（¥¥KGIT01）、および企業のコード集合（¥¥KGCO01）の内容を表示するコマンド（Iは、INFOの省略形）。

企業コードは“証券コード”，項目コードは日本興行銀行“項目コード”による。<sup>(5.8)</sup>

(2. 2) 主成分分析例題（ANAL14：財務比率の総合2：付録2参照）

1 ~ 2 >> LET @NPC = 5; @NPC1 = @NPC+1:

スカラー変数“@NPC”は、入力する係数（比率）の数。

4 >> MAT MD1 = NORD(MMX)';

標準化データ行列関数“NORD”（平均0，標準偏差1）を用いて、本例題“（ANAL13：財務比率の総合1）”で作成した各社の財務比率行列（MMX）の転置行列を標準化データ行列に変換し、そのまた転置行列を行列“MD1”の名前で登録する行列演算式。

(5.8) 項目コードは下記資料を参照せよ。

- ・「興銀財務データファイル項目説明——新フォーマット——」，日本興行銀行・興銀データサービス。
- ・本叢書〈付録3〉「興銀財務データ項目」

転置行列としたのは、行列の次元を入力データ行列 (MMX) に合わせるためである。

5>> MXR = CORX(MD1);

標準化データ行列 (MD1) の相関行列を CORX 関数を用いて求め行列名 “MXR” で登録する。生データ (MMX) を引き数に指定することも出来る。

6>> MXX = EIGNH(MXR);

相関行列 “MXR” の固有値 (m 個) とそれに対応する固有ベクトルを求め行列 (MXX) として登録する。

算出された固有値、および固有ベクトルは固有値の大きい順に並べられ出力される。

入力相関行列 (MXR) は、対称行列であるから組込関数 “EIGNH” (ハウスホルダー法) により求める。

非対称行列の場合は、組込関数 “EIGNQ (QR法)” を用いる。

7>> MXS = SDEV(MD1' \* MXX(/2,@NPC1)); .....①

8>> MW2 = MXX(/2,@NPC1) \* DIAG(1/MXS); .....②

9>> MD2 = MD1' \* MW2; .....③

上の3コマンドにより標準化された第1主成分から第5主成分値が求められ行列名 “MD2” で登録される。

①は、第1主成分から第5主成分値までの標準偏差を組込関数 “SDEV” を用いて求めている。

MXX (/2, @NPC1) は、第1～第5固有値に対応する固有ベクトルを指定している (行列名標で行の添字指定が無い場合は全行を対象とする、列についても同様である)。

②は、第1～第5固有ベクトルを①で求めた主成分値の標準偏差 (MXS) で除し、標準化した固有ベクトル (MW2) を求める。

実際の計算は、MXSベクトルの逆数を組込関数(DIAG)を用いて対角行列にし、それと第1～第5固有ベクトルとの積の形で求めている。

③は、②で求めた固有ベクトル(MW2)を元に標準化した5系列の主成分値を算出している。

1 0>> MST = MW2' \* MXR:

元の変量と主成分との関係を見るため構造係数を第1主成分～第5主成分について計算する。

1 1>> DMX MXR:

それぞれの計算経過をDMX文を用いて出力する。

行列名にアポストロフィーがつけられている場合は転置形で出力される。

1 2>> DMX SUBMX(MXX/1,-5/1,-5/):

SUBMX関数指定によりMXX行列に出力されている固有値、および固有ベクトルを出力する。1列目に固有値の並び、2列～5列はその固有値に対応する固有ベクトルを出力。

行列MXSは、紙面の都合上“1 6>>”で出力する。

1 3>> DMX MW2:

標準化した固有ベクトル(MW2行列:表示の1列～5列)を出力。

1 4>> MD2:

行列“MD2”が目的の主成分値。

1 5>> DMX MST':

行列“MST”が構造係数で、元の変量と主成分との相関関係を表わす。

転置形で出力しており表示の第1列が第1主成分に、行が元の変量に対応している。

1 6>> DMX MXS:

MXSは、主成分(第1～第4)の標準偏差。

1 7>> SCX SUBMX(MST/2,1/1,-5/) VALUE -0.5,1:

元の変量 ( $X_j$   $j=1,5$ ) と主成分との関係を構造図で表わす。

この図では、第1・第2主成分間の関係を表しており、第2主成分 ( $M$  XS行列の第2列) をY軸に、第1主成分をX軸に指定している。

VALUE句を書き、X軸とY軸のスケーリングで最小値を  $-0.5$  , 最大値で1が指定されている。

総資本回転率 (プロット記号1), 売上高営業利益率 (同2), 流動比率 (同3), 当座比率 (同4), 総資本経常利益率 (同5) である。

この例では、入力変数 (財務比率等) が少くサンプル (企業) も少ないため図からの確かな分析・評価を行うことは出来ないが次の現象情報の出力とコメントをすることが出来る。

- (a) プロット記号3 (流動比率) および4 (当座比率) は、第1・第2主成分共プラスの値を示している。両比率共企業の短期的支払い能力を表し企業の安全性を示す指標である。
- (b) プロット記号2 (売上高営業利益率) および5 (総資本経常利益率) は第1主成分 (Y軸) がプラスで第2主成分 (X軸) に対してはマイナスの値を示している。両比率共企業の収益性を表す代表的な指標である。
- (c) プロット記号1 (総資本回転率) は、第1主成分に対して殆ど値を持たず、第2主成分に対してはマイナスの値を示している。総資本回転率は、営業収益をあげるための総資本の利用度を表し総括的な企業経営活動の程度を見る指標として用いられる。

総合判定では、第1主成分が企業の健全性、第2主成分は企業の収益性を表す因子と見られるが、実際の分析に於いては対象とする変量 (比率、成長率等)、ケース (企業) 数を多くし、種々の条件 (変量に関してはその各種セット、ケースに関しては業種・規模等) の下での分析結果を積み上げることが必要である。

1 8>> SCX SUBMX(MD2/1,-10/2,1/):

第1主成分値(X軸)と第2主成分値(Y軸)の関係を各サンプル(企業)について構造図で表わす。データは、標準化データ(平均0,標準偏差1)であるので対象全企業の相対的な関係で表されている。

プロット記号“1”は第1番目企業(分析例ANAL13のGETR文)、“A”は10番目に検索された企業のデータを表す。“ロ”は重複プロット企業。

SCXコマンド同様SCSコマンドでもVALUE句を書くことによって、X軸とY軸のスケーリングを任意の値にかえることができる。

1 9>> IU MD1,MD2,MST,MXR,MXS,MXX,MW2:

この例題で用いた(登録した)行列の諸元を表示する。登録した行列は、削除(DEL USER.行列名)しない限りいつまでも残る。

《記》

- ・SCXコマンドによる構造図のプロットでは、コマンドのオペランドを行列で指定する(第1行・Y軸,第2行・X軸)。
- ・プロット記号に関しては、本叢書〈付録1〉「経済経営情報分析システム(SECRETARY)の記述文法」“SCATTERS(SCS),SCATTERX(SCX)コマンド”の項を参照せよ。

(2.3)主成分分析例題(ANAL15:財務比率の総合3:付録2参照)

例題-ANAL13,ANAL14の計算結果の評価と検証を計算論理面より行う。

但し、このプロシージャ(ANAL15)の実行の前に、“ANAL13”および“ANAL14”の実行により、標準化データ行列・相関行列・固有値、および固有ベクトルが算出されているものとする。

1~2>> MAT MW1 = CUMU(MXX(/1)); .....①

$$MW2 = MW1 / @NPC * 100; \dots\dots\dots\textcircled{2}$$

次式で求められる“累積寄与率”を計算し、各主成分値の説明力を調べる。

$$\sum_{k=1}^m C_k = \frac{(\lambda_1 + \lambda_2 + \dots + \lambda_m)}{p} \quad (m \ 5.28)$$

式 (m 5.25) を行列で表すと、

$$R A = \Lambda A \quad (m \ 5.29)$$

“A” は、直行行列 ( $A' A = I$ ) であるから、

$$\Lambda = A' A R = R \quad (m \ 5.30)$$

この式のトレース (tr) をとる。

Rは相関行列でありそのトレースは次式で表すことが出来る。

$$t r (R) = p \quad (m \ 5.31)$$

したがって、 $\Lambda$  のトレースは、次式で表される。

$$\lambda_1 + \lambda_2 + \dots + \lambda_m = p \quad (m \ 5.32)$$

“p” は変数の数。

①は、M X X 行列の 1 列目、即ち全固有値の並びの累積値を組込関数“CUMU”で計算しMW1の名前で登録する。

②が目的の累積寄与率(%表示)である。pは、既知の値であり定数で与えている。

これらの計算結果の出力は、行番号7のDMX文で表示されている。

この表は行 (ROW) が第 i 主成分に対応し、列については 1 列目が累積固有値の並び、2 列目が累積寄与率(%表示)を表わしている。

第 1 主成分の寄与率は 50.4 で表わされているが、この意味は元の5つの変量による全変動のうちの 50.4 %を第 1 主成分が占めているを示している。

第 1、および第 2 主成分で 87.5 %の説明力を持ちデータの要約が行わ

れていることが確認できる。

3~4>> MX = MXX(/2,@NPC1)' \* MXX(/2,@NPC1); .....①

MW3 = MXX(/2,@NPC1)' \* MXR \* MXX(/2,@NPC1);...②

固有ベクトルを列とする行列 “ $A = [a_1, \dots, a_p]$ ” が直交行列であることを確かめる。

直交行列は、

$$A'A = I \quad (I: \text{単位行列}) \quad (m 5.33)$$

を満足する行列をいい、これはAの逆行列とAの転置行列とが等しい行列であるということもいえる。

計算したMXの値(単位行列)は、行番号8のDMX文で表示されている。

また、固有値が全て異なるか、Rが対称行列であれば、

$$A^{-1}RA = \Lambda \quad (m 5.34)$$

により直交行列を用いて行列 “R” を対角化することができる。

②のコマンドの計算結果(MW3)は、行番号9のDMX文で表示されている。

5~6>> MY1 = MXR \* MXX(/2,@NPC1); .....①

MY2 = MXX(/2,@NPC1) \* DIAG(MXX/1)'; .....②

おわりに、固有値および固有ベクトルが “ $RA = \Lambda A$ ” 式を満足する形で計算されているかを確認する。

①は、 $RA$ の計算。

MXR: 相関行列 ..... (Rに対応)。

MXX(/2,@NPC1): 固有ベクトルを列とする行列 ..... (Aに対応)。

②は、 $\Lambda A$ の計算。

“DIAG” は、行ベクトルを対角行列にする組込関数で固有値の並び(列ベクトル)を転置しDIAGの引き数としている。

①および②の計算結果は、行番号10・11に表示されており同じ値で

あることが確認できる。

## 5. 6 財務分析・経営計画支援総合情報分析システム

財務分析は、企業経営活動の過程・結果から発生・生成されるデータ、および、企業経営の環境から計測され、あるいは調整・制御されるデータ等を用いて、比率分析をはじめとする各種の分析手法を駆使し企業の財務構造の実態を捉え、企業経営の健全性・成長性確保等の意思決定のための情報を企業内と利害関係者に提供することを目的とする。通常は、分析対象の問題領域を狭い範囲に限定した分析情報を多数用意しそれぞれの情報の解析・評価、および全体の総合判断は、それらの分析結果をもとに意思決定者の経験と主観によって行われる。

このような伝統的意思決定の方法・手続きを一步進め、ヨリ多くの情報を総合且つ同時的に捉え分析の自動化をコンピュータを用いて行うという方法もある。

現在のコンピュータの利用環境の下で財務分析の問題を総合的視点で捉え意思決定のための情報提供を行っていくには、大凡次のような問題の設定・記述、処理、制御機能を備えたファシリティー、支援ツールが必要となる。

- (a) 分析対象分野のデータファイル化機能、または利用データが蓄積されているデータベースとのインターフェース。
- (b) データ作成・変容・編集・表示機能。
- (c) 分析モデルの記述・実行機能。
- (d) シミュレーション機能（分析モデル、パラメータの変更・制御等）。

これらの機能を備えた分析システムは 1970 年代中頃に実用システムが開発されている。その代表的なシステムとして、米国 EXECUCOM社の“IFPS (Interactive Financial Planning System)”，米国 SSI (Social Systems, Inc.) の“SIMPLAN (Simulation Program for Corporate

Planning) ” が上げられる。

それぞれ、分析モデル構築・シミュレーション機能、目標探索・危険分析・感度分析等の機能を持つ対話形式のシステムである。

本項では、IFPSを用い財務諸表を入力素材として損益に関する財務項目の予測と収益性の簡単な分析を試みてみることにする。

《記》 実行例は、下記の部分に掲載。

〈付録2〉財務分析問題（第5章）実行例 “【分析例 ANAL16】

IFPSによる財務分析”。

《注記》

- “?” 記号は、IFPSのコマンド・プロンプト。  
? 記号に続く文字列は使用者入力情報。
- 分析プログラム “ANAL16” は、エディタにより予め作成されているものとする。
- 左端の数字は、IFPSの行識別番号（説明では、行番号と呼ぶ）。

(1) 分析モデルのリストと説明（付録2参照）

COLUMNS 0-5

分析（予測）期間を指定する。0は当該年度，5は5年先。

SALES = 2921473, PREVIOUS SALES \* 1.03

売上高 (SALES) を定義する文。“PREVIOUS” は予約語で各計算が予約語の後に書かれている変数のコラム (COLUMNS 文で定義された期) の前の期 (年) のデータに基づいて行われることを意味する。

この文は、各期3%の伸びを示す売上高の時系列データを定義している。

STLINE DEPR(INVESTMENT,0.1,5,DEPRECIATION,BOOK VALUE,  
CUM DEPRE)

定額法により減価償却額を求める関数。

減価償却は、次式によって求められる。

$$\text{年間減価償却額} = (\text{資産原価} \times (1 - \text{残存簿価率})) / \text{耐用年数}$$

当該関数は、資産原価・残存簿価率・耐用年数を入力として指定することにより、期間償却額・簿価・累積償却額が求められる。

DEF = IRR(NET CASH FLOW, INVESTMENT)

資金流列(キャッシュ・フロー)を入力し内部収益率(IRR: Internal Rate of Return)を求める関数。

(2) 分析結果(付録2参照)

(3) 分析モデルの試行テスト1(WHAT IF 機能: 付録2参照)

“WHAT IF”は、分析モデルあるいはデータの一部分を変えた場合全体がどのように変わるかを試行錯誤的に観察する機能である。

? WHAT IF

WHAT IF CASE 1

ENTER STATEMENTS

? SALES = 2950000, PREVIOUS SALES \* 1.028

? SOLVE

ENTER SOLVE OPTIONS

? BEP, NET CASH FLOW

..... 分析結果リスト .....

(4) 分析モデルの試行テスト2(GOAL SEEKING 機能: 付録2参照)

“GOAL SEEKING”は、特定の変数の目標値を定めたときそれを達成するためには他の変数(項目)はどのような値を取るべきかをシステムが解析して出力する。

? GOAL SEEKING

GOAL SEEKING CASE 1

ENTER NAME OF VARIABLE TO BE ADJUSTED TO ACHIEVE

PERFORMANCE

? SALES

ENTER 1 COMPUTATIONAL STATEMENT(S) FOR PERFORMANCE

? GROSSMGN=0

..... 分析リスト .....

ENTER SOLVE OPTIONS

? BEP, NET CASH FLOW

..... 分析結果リスト .....

I F P Sはこれらの試行テストのほか，ある項目の値を段階的に変えた場合他の項目はどう変化するかをみる“SENSITIVITY（感度分析）”，不確実性下の事象即ちリスクを伴った問題に対して確率の概念を導入して分析する“RISK ANALYSIS”等の分析を行うことが出来る。

## 5. 7 財務諸表作成支援システム

経営活動で発生した取引のデータを簿記によって記録整理（仕訳表から総勘定元帳作成，試算表・棚卸し表作成等）しその資料を元に各種の財務諸表が作成される。この作成作業は，財務会計の基礎理論に乗っ取り財務諸表作成時点に於ける会計基準・会計公準・会計原則・関係税法等に準拠した報告をしなければならないという制約を受けること，大量の記録データを整理・統合し各種の財務諸表全てを体系且つ関連財務項目相互の均衡が保たれる形で作成しなければならないという困難な問題を持っている。

このような背景から財務諸表作成関連のソフトウェアは多種にわたり量的にも膨大で通常専用ソフトウェアによって処理されている。

財務諸表を作成する一連のソフトウェアの開発のためには莫大な費用と開発期間を必要とするため会計業務の処理を総合的に行う専用システムがソフトウ

ェア・ハウスで提供されビジネスとしても市場が形成されている。

我々の開発した情報分析システムでの役割は、財務諸表作成の前段階で表全体を概算で捉え、報告される正規の表作成の指針となる情報を取得・提供することにある。

ここでは、連結決算に関する情報分析の一例を示すにとどめる。

(a) 連結決算子会社の検索とその会社コードのカタログ化

(ANAL17: 付録2 参照)

1 >> CFILE NEFA

指定のデータベース (NEFA: 興銀財務データ) をカレント・ファイル (ファイル指定を省略出来るファイル) とする。

2 >> GET 4 TO 5 ACCITEM=117 & STARTY<91 & DATA(91)=1162:

データベースのレコードを条件検索する。“4 TO 5”は、企業コード (証券コード) の頭1桁が4から5迄の企業グループの指定。

データ項目の91番目の欄 (DATA(91)) には親会社の会社コードが入れている。

(b) 親会社に対する子会社の構成比の計算出力 (ANAL18: 付録2 参照)

《記》

カタログ (名前の頭に “¥¥” 記号) の内容は、次の例 (ANAL19) を参照せよ。

3 >> GETR ¥¥REN03I REPEAT ¥¥REN03K:

カタログされている会社名リスト (¥¥REN03K) と項目リスト (¥¥REN03I) によりデータレコードを検索する。

5 >> @ANS=@COUNT;

スカラー変数 “@COUNT” には、その時点の検索レコード数が入る。

7 >> @NOK=NOWD(¥¥REN03K);

“NOWD” は、カッコ内に指定するカタログの内容の語数をカウン

トする組込関数。

12 >> DO @KM=1 TO @NOI:

END文 (16 >>) 迄の文を検索項目数 (@NOI) 回繰り返し実行する。

15 >> #(@ANS) = #(@KS) / #(@KM):

親会社の財務項目 (売上高～当期利益) に対する指定子会社 (@CONO) の対応する財務項目の比率を求める。

“#” 記号はアクティブ・エリアを示す。@ANS は結果の記憶域、@KS は子会社の財務項目、@KM は親会社の財務項目それぞれの位置を指し示すスカラー変数。

17 >> DTM # (1) TO #(@NOI):

親会社の財務各項目のデータ表示。

(c) 親会社に対する子会社の構成比の計算計算 (ANAL18 続き)

(ANAL19: 付録2 参照)

1 >> DTM #(KS1) TO #(@KS), #(@ANS1) TO #(@ANS):

子会社の財務各項目のデータ表示と、親会社に対する子会社の構成比を表示する。

2 >> INFO ¥¥REN03:

¥¥REN03 関係のカタログ内容表示。

3 >> D SCAL @NOI,@NOK,@COUNT:

スカラー変数の値を表示。

(d) 単独決算/連結決算対応表出力 (ANAL20)

同一企業の単独決算と連結決算の財務項目を同時に検索し比較する。

1~2 >> FILE KN1A: CFILE NEFA:

“KN1A”は、単独決算データの収録されたデータベース，“NEFA”は連結決算のデータベース。

CFILE 文はカレント・ファイルとして定義される。

5 >> GET KN1A.6502311000 ..... ADD;

“KN1A”はカレント・ファイルでないため項目コードをファイル名で修飾する。“ADD”は追加モードで検索するためのキーワード。

15 >> DTM # (1) TO # (5):

連結決算財務各項目のデータ表示。

(e) 単独決算・売上高の表／連結決算の比率表 (ANAL20 続き)

16 >> DTM # (6) TO # (15):

単独決算財務各項目のデータ表示と、連結決算との比率の表示。

## 5. 8 その他の分析 (ポートフォリオ, CAPM等)

その他、財務関連の分析は財務理論の発展と広がりに対応し、投資問題とそれに関連するポートフォリオ理論に基づく分析、CAPM (資本資産評価モデル: Capital Asset Pricing Model), MM理論 (Modigliani F. と Miller M. 共著「資本コスト, 企業金融, および投資理論」) に基づく企業評価, 配当政策と利益・株価等の関係分析等々財務分析の問題領域は広大である。<sup>(5.9)</sup>

---

(5.9) これらの問題は筆者が論じる範囲をこえているので研究者・専門家の論文, たとえばつぎの論文を参照されたい。

- ・山地秀俊, 「SECRETARYによるファイナンス理論の検証」, 経済経営研究叢書, 経営機械化シリーズ 19, 神戸大学経済経営研究所, 1986, pp.327-348。
- ・山地秀俊, 「企業の資産選択行動に関する実証的研究」, ——SECRETARYによるLESの推定——, 経済経営研究叢書, 経営機械化シリーズ 19, 神戸大学経済経営研究所, 1986, pp.349-389。



## 《 付 録 》

### 〈付録 1〉 経済経営情報分析システム (SECRETARY) の記述文法

付録1目次	頁
付1. 1 基本(共通)事項 .....	168
付1. 2 名標 .....	168
付1. 3 機能別命令文(コマンド) .....	175
付1. 4 命令文の詳細(アルファベット順) .....	177
付1. 5 行列処理関係 .....	234
付1. 5. 1 行列名標の書き方 .....	234
付1. 5. 2 小行列関数(SUBMX)について .....	236
付1. 5. 3 固有値及び固有ベクトルを求める関数 .....	237
付1. 6 本情報分析システムと統計データベース .....	238
付1. 7 SECRETARY命令文サマリー .....	241

#### 経済経営情報分析システム (SECRETARY) の文法規約

SECRETARYの言語構造については、“1. 3「SECRETARYの構文記述」”で超言語(メタ言語)を用いて示したが、ここでは言語の命令文(ステートメント)の機能・定義・書き方について命令文相互の関連をとりながら説明を含めて記述する。

本情報分析システムは、英・数・記号の組み合わせでつくられる記号列とその集合に独自の約束(文法)を規定し経済経営分野の分析用プログラミング言

語として体型化したものである。設計コンセプトは、対象領域の分析論理に整合した文法を採用すると共に分析者の思考と分析言語の論理ギャップを少なくし且つ分析能率を上げ得る問題記述言語を目指した。

### 付 1. 1 基本（共通）事項

#### (1) 自由欄形式（フリーフォーマット）

命令文の入力欄は特に定められていないので入力域の何処へでも書くことが出来る。入力情報の単語相互は、スペース・コンマ・特殊文字（#, @, ¥ を除く）で区切られる。

#### (2) 命令文の単位と送信単位

命令文の区切りはコロン（:）による。命令文がセミコロン（;）で終わる場合は直前の命令文と同種のコマンドとみなし機能部を省略することが出来る。

送信単位とは、送信キー（エンターキー）を押し命令文、またはその集合を処理系に送ることを言う。1送信単位に複数個の命令文を含む場合、それを命令文集合（コマンドプロシージャ）と呼ぶことにする。送信単位のコロンは省略することが出来る。

#### (3) コメント（注釈）

FORTRAN, PL/I 言語にある注釈文はないが“NE文（省略形）”でそれに代えることが出来る。

### 付 1. 2 名 標

本分析システムでは、データベースをはじめ分析システム自身も数種のファイルと記憶空間より構成されており、それぞれの名前を特定する識別名が必要

である。その識別名を名標と呼ぶことにする。

(1) 変数名

変数名は、分析システムで最もよく用いられる名標の一つであり次のものに分けられる。

- スカラー変数： 第1文字が@マークで、12文字以内の文字列。
- ベクトル変数： 第1文字が英字で、12文字以内の文字列。
- 行列変数： 第1文字が英字で、7文字以内の文字列。

(行列変数は、行ベクトル、列ベクトルを含む)

これらの変数のデータ形式は、浮動小数点方式で10進7桁の精度を持つ。変数名は、使用者の定義するものと分析系命令文の実行結果としてシステムが定義(第1文字“¥”)するものに分けられる。

次の表は、変数定義および変数参照と主要なコマンドとの関係を示す。

変数区分	定義(又は削除)コマンド	参照コマンド
スカラー変数	LET, COM, MAT	LET, COM, MAT, PD, ROW
ベクトル変数	LET, COM, PB, AGK, EST, U, DEL, 分析系コマンド	LET, COM, 情報出力, 分析系コマンド
行列変数	MAT, PU, MU, DEL	MAT, GU, ML, DMX, DMY, 分析系コマンド(SUBMX関数による引用)

《記》コマンドは、省略形で表した。

### ◎ベクトル変数の書き方

ベクトル変数は、オプションとして添字を書くことが出来る。それによってベクトルの成分、時系列データのラグ（又はリード）等を表すことが出来る。

ベクトル変数名 ( $j_s$ ,  $j_t$ ,  $j_l$ )

- $j_s$  (列開始),  $j_t$  (列終端),  $j_l$  (列変位) は,

データ並びの位置を示す指標で定数, スカラー変数, または単純添字式 (スカラー変数 {+-\*} 定数) を書くことが出来る。

{ } 内は, 何れかを選択。

- 添字は, 全てオプション指定。但し, 位置パラメータのためコンマは省略することは出来ない。

《時系列データのラグ指定の例》

CPI変数の1期ラグ:  $CPI(, -1)$

### ◎行列変数名の書き方

行列に関しては, 付1.5「行列処理関係」で触れる。

#### (2) 定数

定数は, 次のものに分けられる。

- (a) 数値定数: LET, COM, MATRIX (MAT) 文の左辺に書くことで定義されシステムテーブルに登録される。

SECRETARYのセッション中有効である。

- (b) 文字定数, リテラル: 文字定数は, 第1文字が英字で12文字以内の文字列。リテラルは, アポストロフィー ( ' ) で囲まれた文字列 (60文字以内)。

#### (3) カタログ名

カタログ・ファイルに登録された名前。文字列情報で通常次のものが入られる。

- (a) 命令文集合（カタログ・プロシージャ）。
- (b) 統計コード，会社コード，表の項目名等のリスト。

カタログ・ファイルへ登録された命令文集合は，その登録名を処理系へ送信することにより命令文が連続して実行される。

また，分析で用いられるデータをコードの集合として登録しておくことにより，検索系コマンド（GET，DBL等）でそのコード集合を引用することが出来る。カタログへの登録は，SAVE（又はS），UC，PC等のコマンドで行われ，INFO（又はI）コマンドで内容を見ることが出来る。

#### (4) ファイル名（データベース名），索引名，統計コード

本分析システムは統計データベースとリンクされ，CFILE文，またはFILE文で指定データベースを動的に定義することが出来る。

《記》統計データベースは，国内外のマクロ経済統計，企業財務マイクロ・データ等の時系列データで現在の収録は400万系列を越える。

この分析システムで用いるファイル名は4文字のデータベースID名を指し，索引名は作成済みの索引ファイル名，統計コードはデータベース毎に定めているレコードの識別名。

#### (5) フィールド名

データベースの情報を参照する最小単位は欄（フィールド）である。分析システムが検索系コマンドを用いてデータベースより受け取るデータは物理データであるため，分析システムとデータベースとの対応関係を記述し名前によってデータを参照できるようにしておくことが必要である。この関係記述で定義された分析システム側の論理的名前をフィールド名と呼ぶ。関係記述は，使用者が作成・編集することの出来るファイルで，使用者がデータベースのレコードに対して独自のフィールド名を定義することも出来る。定義情報は，DISPLAY（又はD）で確認することが出来る。

## (6) #表示

“#”は、アクティブ・エリア (ACTIVE) のデータを指定するときを用いる代名詞。次に示すようにマトリックス、ベクトル、またはスカラーを表すことが出来る。

# (i<sub>s</sub>, i<sub>t</sub>, i<sub>1</sub>/j<sub>s</sub>, j<sub>t</sub>, j<sub>1</sub>)

- i<sub>s</sub> (行開始), i<sub>t</sub> (行終端), i<sub>1</sub> (行変位) および,
- j<sub>s</sub> (列開始), j<sub>t</sub> (列終端), j<sub>1</sub> (列変位) は、  
行、または列の位置を示す指標で定数、スカラー変数、または、単純添字式 (スカラー変数 {+-\*} 定数) を書くことが出来る。
- 添字は、全てオプション指定。但し、位置パラメータのため行、または、列でオプション指定が1つ以上あるときはコンマを省略出来ない。
- 書き方は、行列名に準じる。

## (7) 関数名

関数は、配列演算文 (LET, COMPUTE 又は COM), または、行列演算文 (MATRIX, 又は MAT) で用いられる。

関数は、初等関数と行列関数に分けられ、前者は、LET文、COM文、MAT文で用いることができ、後者は、MAT文でのみ用い得る。

但し、EPA関数はMAT文で引用することは出来ない。

(a) 初等関数

関数名と書き方	機 能
ABS (v)	ベクトルv (以下同様) の絶対値
COS (v)	余 弦
CUMU (v)	累 積 値
EPA (v)	季節調整 (EPA法)
EXP (v)	指 数
EXPECT (v,p)	期待値 (v : データ, p : 確率)
LOG (v)	自然対数
LOG10 (v)	常用対数
MEAN (v)	平均値 (出力 : スカラー)
NOWB (v)	レコード内 (ベースエリア) 語数
NOWD (v)	レコード内 (カタログファイル) 語数
SDEV (v)	標準偏差
SIN (v)	正 弦
SQRT (v)	平方根
TAN (v)	正 接
TBLN (s)	正規分布累積確率
TOTAL (v)	合計値 (出力 : スカラー)

《記》 v : ベクトル。

s : スカラー。

## (b) 行列関数

関数名と書き方	機 能
x' 又は (x)'	転置行列 [TRANS (x) でも可]
x" 又は (x)"	逆行列 [INV (x) でも可]
ABS (x)	行列 x (以下同様) の絶対値
CORX (x)	相 関 行 列
COS (x)	余 弦
COVX (x)	分散共分散行列
CUMU (x)	累 積 値
DET (x)	行列式 (スカラー: x 行列の行列式)
DEVX (x)	偏差積和行列
DIAG (x)	対角行列 (行ベクトルを対角行列に変換)
EIGNH (x)	固有値及び固有ベクトル (実対称行列)
EIGNQ (x)	固有値及び固有ベクトル (実非対称行列)
EXP (x)	指 数
IDENT (n)	単位行列 (n は行列のサイズ)
LOG (x)	自然対数
LOG10 (x)	常用対数
MEAN (x)	平均値 (出力: ベクトル)
MULT(x, y)	行列 x, y の対応する要素の積)
NORD (x)	標準化データ行列
PRODX (x)	積 和 行 列
SDEV (x)	標準偏差 (列の標準偏差を求める)
SIN (x)	正 弦

SIZE (x)	行列サイズ (出力: ベクトル (1×2))
SQRT (x)	平方根
SUBMX (i)	小行列 (小行列関数を参照せよ)
SUM (x)	全要素和 (出力: スカラー)
SUMX (x)	行和 (出力: ベクトル)
SUMY (x)	列和 (出力: ベクトル)
TAN (x)	正接
TOTAL (x)	合計値 (出力: ベクトル)
TRACE (x)	x のトレース (対角要素の和)

《記》

- v はベクトル, x は行列, n は定数またはスカラー変数。
- i は, 小行列関数 (SUBMX) の添字並び (リスト)。
- 出力がスカラーとなる関数: DET, SUM, TRACE。
- 出力がベクトルとなる関数: MEAN, SDEV, SIZE, SUMX, SUMY。
- SUBMX は, 添字並び (リスト) により, スカラー, ベクトル, マトリックスの何れかになる。
- EIGNH 及び EIGNQ 関数は, 左辺に書いた行列変数に特別の配置で出力される (図付 1. 5. 3 固有値・固有ベクトルの配置を参照せよ)。

付 1. 3 機能別命令文 (コマンド)

(1) 制御及び共通命令文

CALL, DISPLAY(D), IF, HELP(H), INFO(I)-INFOBASE(IB)-INFOUSER (IU), DO-END, DO WHILE-END, MODE, OVER(または, /\*)

## (2) 定義文

CFILE-FILE, FIELD(F), LISTOPTION(LOP), NEWCON-NEWCOFF,  
PERIOD(PD)-ROW-COL

## (3) データの流れを制御する文

GET(G)-DBLIST(DBL), GETREPEAT(GETR)-GETRR, GETCROSS  
(GETC)-PUTCROSS(PUTC), GETRR, TRANSPOSE(TR), PUTBASE  
(PB)-GETBASE(GB), PUTUSER(PU)-GETUSER(GU), GETPUTC  
(GETP), OPEN, OPENTMPF, READTMPF(RDT)-READSUM(RDS),  
PUTCATALOG(PC), READMT(RDM)-WRITEMT(WTM)

## (4) ストック（記憶空間）の制御文

AGK-AGM-AGS, CHANGEDATE(CD), CLEARACTIVE(CA), DELETE  
(DEL), SAVE(S), UPDATECAT(UC), RENAME(REN), SORT,  
UPDATE(U)

## (5) 情報の出力文

CHANGECODE(CC), CHANGETITLE(CT), DATA(DT)-DATAG(DTG),  
DATACROSS(DTC), DTCPLOT, DATAMPL(DTM), LINE, NOPAGE,  
OUTDEV, PAGE(PG), PLOT(PL), SCATTER(SC)-SCATTERN(SCN),  
SKIP, STATIS(ST), TITLE(T)

## (6) 演算及び分析文

LET-COMPUTE(COM), ESTIMATE(EST), ESTY-ESTYS, LSM-LSMS-  
LSMC-LSMCS-LSMN-LSMNS, MOVINGAV(MAV), RANDUNIF(RNU)-  
RANDNOR(RNN)-RANDEXP(RNE)-RANDPOIS(RNP), SUMMATION  
(SUM), TREND(TRD)

## (7) 行列処理関係

MATRIX(MAT), MXUPDATE(MU),DELETE(DEL), MXLIST(ML),  
DMX, SCATTERS(SCS)-SCATTERX(SCX)

(8) その他の命令文

NOEFECT(NE)

付1. 4 命令文の詳細 (アルファベット順)

《注記》

- ・コマンド名に続く括弧内の文字列は省略形コマンド (例: GET(G))。
- ・ハイフン (“-”) で区切っているコマンドは、関連コマンドであることを表す (例: GET(G)-DBLIST(DBL))
- ・各コマンドの識別のために、コマンド項目の頭に “>” 記号を付ける

《表記法に付いて》

- ・ “ [……] ” で囲まれた句は、オプション指定 (省略可)。
- ・ “ {……} ” で囲まれた句は、その内の一つを選択する。

>> AGS, AGM, AGK

時系列データの期種変更

AGS [識別子A = ] 識別子B [TO 目的期種]

[PREFIX 前置文字列]

時系列データの期種変更 (月次→四半期・年次, 四半期→年次) を行う。TO 句の初期設定は、月次データならば四半期 (Q), 四半期データならば年次 (A) が仮定される。

PREFIX 句の指定をすると”識別子A=”を省略できる。その時は識別子Bの頭にcの文字列を付けて登録する。TO 及びPREFIX 句の情報は、次に同じコマンドで指定するまでシステム情報として保存される。

- ・コマンド: AGS (累算), AGM (単純平均), AGK (残高)。
- ・識別子: ベースエリアへの登録名, または, # (i) ・ # Y (i) 表示。

i は、自然数（詳しくは、LET コマンドの説明を参照せよ）。

- 目的期種等： Q（四半期）、A（年次）、または自然数。

自然数を指定した場合は、原系列をそのインターバルで編集する。

- 前置文字列： 3文字迄の文字列。

識別子B の頭に付けられる（初期設定は ¥）。

### <使用例>

- ① AGS QDM = DM

月次データ (DM) を四半期データ (QDM) に変換する。

- ② AGM DM TO A PREFIX B

月次 (DM) を年次に変換 (単純平均) し変数名の頭に "A" を付加した名前 (BDM) でベース・エリアに登録する。

### >> CALL

命令文集合の実行

#### [CALL] プロシージャ名 [C]

登録済みの命令文集合 (コマンド・プロシージャ) を呼び出して実行する。

命令文集合に修正を施して実行させる場合は、"C"を指定する。

CALL文は、多重呼び出し(nesting) ができるので各種の機能を体系的に整備しそれらを組み合わせて使うことにより分析の能率を上げていくことができる。

CALL は省略可能でプロシージャ名のみで実行することもできる。

但し、SECRETARY コマンドと同じプロシージャ名の場合は、CALL を必要とする。

- プロシージャ名： SAVE,または UC コマンドで登録された名前。
- 関連コマンド：

INFO ; プロシージャ名等カタログの一覧表表示。

SAVE ; コマンド・プロシージャを登録する。

UC ; コマンド・プロシージャを更新する。

DEL ; コマンド・プロシージャを削除する。

REN ; コマンド・プロシージャの名称を変える。

〈使用例〉

① CALL EX1

EX1 を呼び出して実行する。

② EX2 C

EX2 に修正を施して実行する場合。

>> CFILE

データベースの定義, または, 切り替え

**CFILE データベース名**

CFILE 文は, カレント・ファイルとして定義するときに用い, カレント・ファイルのレコード検索の場合, 統計コードの指定のみで検索できる。

・関連文: FILE

定義したデータベースの情報は, "D DB" でみられる。

・データベース名: 主要なものを下記に示す。

IFS (国際資金), BOP (国際収支), NES (日経総合経済),

KG1A (興銀財務), CSSA (米国財務), FJTA (米国経済),

ABS (豪州), OEC (OECDデータ), WSTA (世界銀行),

WDTA (世界銀行負債) 等。

3文字の場合は, A (年次), Q (四半期), M (月次) を4文字目に入れる。

〈使用例〉

① CFILE IFSQ

国際資金データベース (IFSQ) をカレント・ファイルとする。

## ② CFILE CSSA

カレント・ファイルを CSSA (S&P 社 COMPUSTAT) に切り換える。

## &gt;&gt; CHANGECODE(又は CC), CHANGETITLE(又は CT)

画面表示, または印刷時に表示コード, タイトルを変更する。

(a) CHANGECODE (又は CC) 名前 (i) [名前 (i), ……]

(b) CHANGETITLE (又は CT) i '文字列'

(a) は, レコードの表示コードを指定した名前 (12文字まで) にする。

同コマンドを用いない場合は系列コードを表示する。

(b) は, レコードの表示タイトルを指定したものに変わる。

• i : アクティブ・エリアのレコード番号。

## 《条件・制限》

• 文字列は, 60文字まで。

## &lt;使用例&gt;

## ① CC AB(2), CDEF(4)

アクティブの2番・4番のレコードの表示を AB 及び CDEF とする。

## ② CT 3 ' KOBE UNIV.'

アクティブの3番目のレコードのタイトルを "KOBE UNIV." とする。

## &gt;&gt; CHANGEDATE (又は CD)

時系列データの開始時点の変更

CD 名前 [終了名前] 西暦年.期

CD # 西暦年.期

データの収録開始時点を変更する時に用いる。

収録開始時点は, 繰上げ繰下げ何れでも可能である。データが充足しない場合は "0" がうめられる。

- ・名前： ベース・エリア名 (LET 文の左辺で定義, U コマンド等で作成)。
- ・終了名前： 区間に渡って変更する場合に指定する。この場合はこの操作を行ってよいかを使用者に質すので, よい場合は送信キー, コマンドを取消す場合は C を含めてなんらかの文字を送信する。
- ・#： アクティブ・エリアの全てのレコードを対象とする場合。
- ・西暦年.期： 年次データ以外で年のみ指定した場合は1期, または1月を仮定する。期・月は, 83.01, 83.02, …… の書き方をする。

① CD CODE1 81.03

CODE1 の収録開始時点を81年3月に変更する。

② CD NS NT 78

NS から NT までのレコードの収録開始時点を1978年に変更する。

>> CLEARACTIVE (又は CA)

アクティブ・エリア (ACTIVE) のクリアー

CLEARACTIVE(CA)

アクティブ・エリアをクリアーし検索前の状態にする。

>> COL コマンド

列範囲の定義

COL [開始列] [TO 終了列]

行列データを作成するときの列サイズの定義。

或は, 行列の出力 (DMX コマンド) の区間を指定する。

- ・開始列・終了列： 自然数。

<使用例>

① COL 1 TO 15

1列から15列までとして定義する。

## ② COL TO 20

終了列のみ変える。

## ③ COL @START TO @END

予め定義されているスカラー変数で与える。

## &gt;&gt; データ出力系コマンド (その1)

(a) コマンド [識別子, ..., 識別子] [TO 識別子] [識別子, ...]

[DECIMAL p] [PAGE] [LP] [SKIP n]

(b) コマンド 小行列関数 [DECIMAL p] [PAGE] [LP] [SKIP n]

・コマンド (括弧内は省略形) :

DATA(DT), DATAG(DTG), DATAMPL(DTM),

PLOT(PL), SCATTER(SC), SCATTERN(SCN)

(a) は、ベースエリア、またはアクティブ・エリアのデータを各種の出力様式で画面、或はハードコピーに出力する。

(b) は、行列 (又は段落) のデータを出力する時に用いる。

この場合データの収録開始点を1と見なして小行列関数に指標を与える。

データ出力系コマンドには、この外クロスセクション出力を行う "DATACROSS" 簡易統計表の出力を行う "STATIS" 等がある。

・識別子 : ① ベース・エリアの変数名。

② #(i): アクティブ・エリアの i 行目のレコードの指定。

③ #Y(j): アクティブ・エリアの j 列目のレコード。

・小行列関数 : 小行列関数 (SUBMX) を参照せよ。

・p: 出力データ表示の小数点以下桁数 (自然数)。

・PAGE (改ページ), LP (ハードコピー出力), SKIP (改行)。

## &lt;使用例&gt;

## ① DTM

アクティブ・エリアの全レコードのデータを表示する。

② DT B1 TO B5 LP PAGE

ベースエリアの B1 から B5 までのレコードを改ページして LP に出力。

③ DTM B1 # (3) TO # (5) B2

アクティブ・エリアとベース・エリアのレコードの混在出力。

③ PLOT SUBMX (MX/1, -9/2, 5/)

行列 MX の1～9行, 2・5列のデータをプロットする。

## >> データ出力系コマンド (その2)

コマンド [BASIS 基準項目] [CONTROL 欄名]

[DECIMAL p] [PAGE] [LP] [SKIP n]

アクティブ・エリアのレコードを入力としクロスセクション表形式でのデータの出力と、基準項目に対する比率を出力するコマンド。

また、CONTROL 句を指定してサブトータル出力を行うことが出来る。

データ出力系コマンドには、この外各種の出力様式で画面に出力を行うコマンド (その1) がある。

- ・コマンド (括弧内は省略形) : DATACROSS(DTC), DTCPLOT
- ・基準項目: ① ベースエリアの変数名。
  - ② # (i); アクティブ・エリアの i 行目のレコード指定。
  - ③ # Y (j); アクティブ・エリアの j 列目のレコード指定。
 省略時は最初のレコードを基準項目とみなす。
- ・欄名 (フィールド名) : "D I" で確認できる。
- ・p: 出力データの小数点表示 (自然数)。
- ・PAGE (改ページ), LP (ハードコピー出力), SKIP (改行)。

### <使用例>

① DATACROSS

アクティブ・エリアの全レコードのクロスセクション表示。

## ② DTC BASIS BASED1

比率の基準項目を "BASED1" とする。

## >> DBLIST (又は DBL)

データベースのレコード検索

DBLIST コード [コード, …… , コード] [TO コード]

[索引名 関係子 値, …… , 値]

[接続子 条件句] [, 接続子 条件句] [ADD] [LP]

《記》

条件句: 索引名 関係子 値 [値, …… , 値]

DBLIST は、データベースよりレコードを検索すると共にそのレコードの項名目、タイトル、収録期間の一覧表を画面、またはハードコピーに出力する。

当該コマンドは、指定コードの上位概念によって検索が行われる。

名条件式を書けば自然語によるサーチを行うこともできる。

このコマンドを用いることによりデータベースの収録レコードの最新の情報が見られる。

- コード: データベースの系列コード。  
頭2文字が "¥¥" のカタログ名 (コード・リスト) も指定可。
- フィールド (欄) 名, 又は索引名:  
"DISPLAY (又は D) I" コマンドで確認可。
- 関係子: =, ^ =, >, >=, <, <=, 又は ~ (区間指定)。
- 値: 数字, 又は文字列 (12文字以上はアボトロフィーで囲む)。
- TO (区間指定), ADD (追加モード), LP (プリンター) はキーワード。
- 関連コマンド: GET
- コマンドの詳細は, GET コマンドの項を参照せよ。

〈DBL コマンドの使用例〉

① DBLIST

アクティブ・エリアのレコードの一覧表。

② DBL 1119 LP ..... IFS データベース検索  
米国の国民所得関係項目をハードコピーに出力。

③ DBL C1 TO C2 NAME=GM,CHRYSLER ..... CSSA データベース検索  
ベースキー (C1, C2) とレコードの内容による検索。  
NAME は欄名で "D I" コマンドで確認できる。

>> DELETE (又は DEL)

登録済みのカタログ (コマンド集合, コード・リスト) 等の削除

(a) DELETE カタログ名

(b) DELETE BASE.名前B (TO 名前B)

(c) DELETE USER.段落名

(a) は, コマンド・プロシージャ等のカタログを削除する。

(b) は, ベース・エリアのレコードを削除する。

(c) は, ユーザファイルの段落, または行列を削除する。

- ・カタログ名: コマンド・プロシージャ等。
- ・名前B: ベース・エリアのレコード名。TO 句を書くと同間削除。
- ・段落名: ユーザ・ファイルの段落名, または行列名。
- ・関連コマンド: INFO,IB,IU; カタログ等の一覧表。

〈使用例〉

① DELETE PROC1

カタログ (プロシージャ) 名 "PROC1" を削除する。

② DEL BASE. CODEB

ベース・エリアの "CODEB" を削除する。

## ③ DEL USER.TABLE1

ユーザファイルの "TABLE1" を削除する。

## &gt;&gt; DISPLAY

システム状態表示コマンド

## DISPLAY (又は D) 出力指定

SECRETARY のシステム・ステータス (状態) を表示させる。

## &lt;出力指定&gt;

- AC : カレント・ファイルの属性, レコード数等。
- AGGR : アグリゲート・コマンド (AGS 等) の情報。
- COMD : コマンド一覧表。
- DATE : 日付と時間を表示。
- DB : 定義済みのデータベースの一覧表。
- FIELD (又は F) : 投入した FIELD コマンドの最新のもの。
- ITEM (又は I) : 定義されているフィールド (欄) に関する情報。  
索引を持つ欄には, INDEX 欄に索引の一連番号が入れられる。
- KEYWORD : キーワード一覧。
- MENU : メニューの内容表示 (MODE コマンドを入れた場合のみ)。
- SCAL [名前] : スカラーの値を表示。  
名前を指定するとそのスカラー変数の値のみ表示する。
- TG : アクティブ・エリアの最初と最後のレコードの表示。
- SORT : 送信した SORT コマンドの最新の文。

## &lt;使用例&gt;

## ① DISPLAY DB

CFILE又は, FILE文で定義したデータベースの一覧表を表示。

## ② D SCAL @K

スカラー変数 "@K" の値を表示する。

## >> DMX

行列データの出力 (その2)

DMX 行列名 [行列名, ……., 行列名] [TO 行列名]  
 [小行列関数]  
 [DECIMAL p] [LP] [PAGE]

行列データを出力する命令文。行列を複数個指定すると、それらの行列を結合した形で出力する。TO 句を用いると区間にわたって多数の行列を結合して出力する。

結合は、通常列方向に結合される。行方向に結合する場合は、転置行列（行列名にアポストロフィーを付ける）で指定する。

行列自身が表頭・表側情報を持っている場合は表の行および列の識別欄に組み入れられ、持たない場合は、行は "ROW:"，列は "COL:" で表示される。

・行列名：MATRIX 文の左辺、或は PU 文で登録した行列、またはベクトル (IU 行列名 [TO 行列名] で確認できる)。

転置形で出力する場合は、行列名の後にアポストロフィー (') を付ける。

### <使用例>

#### ① DMX

アクティブ・エリアの全データを表示する。

#### ② DMX SUBMX(MX/3,-5/1,-8/)

行列 MX の3~5行、1~8列の小行列を転置して出力する。

#### ③ DMX MX1', YVECT

行列 MX1 (転置形) とベクトル YVECT を行で結合して出力。

## &gt;&gt; DO

繰り返しの制御

(a) DO @v1 = ns TO nt [BY np] :

(b) DO WHILE(@v2 op c):

(C) END [文字列] :

(a) DO 文から END 文までの命令文が、DO 文で指定した条件に従って繰り返される。

即ち、@v1 (制御変数) が ns (初期値) に始まり np 刻み (省略時は 1) で nt に至るまで DO 文以後の命令群が繰り返される。

(b) WHILE 句は、カッコ内にその条件を指定し、それが“真”の間中繰り返される。

WHILE 句は、(a)の場合にも書くことができる。その場合は WHILE 句の指定が優先する。

• @v1, @v2: スカラー変数。

また @v2 の代わりに "FLAG(i)" も指定できる。

(例 FLAG(1) = 9 は TMPF ファイルの終わりを検出したとき“真”)

• ns, nt, np, c: 定数またはスカラー変数。

• op (関係子) : =, ^ =, >, >=, <, <=

## &lt;使用例&gt;

① DO @K=1 TO 10: …… END:

スカラー変数 "@K" が1から10まで1刻みの繰り返し。

② DO WHILE(FLAG(1)=0): …… END:

TMPF ファイルのレコード読み取りが終わりになるまで処理を繰り返す。

TMPF ファイルが終わりになると "FLAG(1)" が "1" に設定される。

関連文を含めた使用例は、“OPEN” 文を参照せよ。

>> ESTIMATE: 最小自乗推定による予測

コマンド [識別子, …] (TO 識別子) (WITH ベース名)

コマンド 小行列関数 (PAGE) (LP)

最小自乗推定コマンド (LSM 及び LSMS を除く) を実行すると, その都度回帰係数をベースエリアの ¥COEF に出力する (LET 文を用いて ¥COEF の情報を別の名前でベースエリアに登録しておくこともできる)。

当該コマンドは, この ¥COEF の情報を基に標本回帰線を求める。

WITH 句でベース名を書いた場合は, そのモデルで推定値の計算を行う。

また, 最小自乗推定コマンド同様, 推定値, 残差は, それぞれ ¥EST, ¥RESI の名前でベースエリアにコマンド実行の都度出力される。

- コマンド: ① ESTS (プロット出力無)。  
② ESTIMATE, 又は EST (プロット出力有)。
- 識別子 (最初に指定したものが被説明変数) :  
① ベースエリアの変数名 (ベース名)。  
② # (i); アクティブ・エリアの i 行目のレコードの指定。  
③ # Y (j); アクティブ・エリアの j 列目のレコード。

《条件・制限》

変数の数は, 推定時と同じにしなければならない。

<使用例>

① EST # (2) # (1)

直前の最小自乗分析で出力された回帰係数を基に標本回帰線を求める。

② EST CY CX WITH COEF1

"COEF1" モデルにより CY の外挿を行う。

>> ESTY (又は ESTYS) コマンド

ESTY 小行列関数1, 小行列関数2 (PAGE) (LP)

パラメータ推定用のデータ行列（小行列関数1）、および被予測データ行列（小行列関数2）の2組のデータ行列を与え、構造と予測に分けて単純回帰モデル（第1列説明変数）により分析を行うコマンドである。

情報は、下記のもので出力される：

(1) 推定値と残差のプロット（ESTYの場合）。

(2) ファイルへの出力として；

① ¥RESIX [小行列関数2の残差行列（1列と他列との組み合わせ）]。

② ¥RESIX1, ¥RESIX2（①の列和ベクトル，後者は絶対値）。

③ ¥RESIY1, ¥RESIY2（①の行和ベクトル，後者は絶対値）。

①はユーザファイルへ，②および③はベース・エリアへ出力される。

- ・小行列関数1：パラメータ推定用のデータ行列（第1列が説明変数）。
- ・小行列関数2：被予測データ行列（列サイズは推定行列に合わせる）。
- ・小行列関数の書き方：4.2.7 (2) を参照せよ。

## >> FIELD (又は F)

表示欄の定義

FIELD 欄名 [, 欄名, ..., 欄名]

DATA, DTM, PLOT, LIST 等の出力系コマンドで表示する欄の定義をする。

FIELD コマンドは次に同コマンドを入れるまで有効である。

欄の初期設定は、レコードのタイトル部分の58文字が設定されている。

- ・欄名： "DISPLAY (又は D) I" コマンドで確認可。

欄名の代わりに "p,1 (エル)" の組としても指定可能である。

p (オフセットポイント値), 1 (欄の長さ)。

<使用例: IFS データベースの場合>

- ① F TEXT

出力系コマンドで "TEXT" 欄を表示欄とする。

② F 1,58

オフセット・ポイント値と、欄の長さで指定した場合。

>> FILE

データベースの定義

FILE データベース名 [ , データベース名, ……]

複数のデータベースを同時にアクセスする場合はこのコマンドを用いる。

FILE 文で定義した場合は、統計コードをデータベース名で修飾しなければならない(例: IFSA.15864)。

・関連文: CFILE

定義したデータベースの情報は、"D DB" でみられる。

・データベース名: CFILE 文を参照せよ。

<使用例>

① FILE KG1A,CSSA

KG1A (興銀財務一部上場) と CSSA (S&P COMPUSTAT) を定義する。

>> GET (又は G), DBLIST (又は DBL)

データベースのレコード検索

GET コード [コード, …… , コード] (TO コード)

DBLIST [索引名 関係子 値, …… , 値]

[接続子 条件句] [, 接続子 条件句] [ADD] [LP]

《記》

条件句: 索引名 関係子 値 [値, …… , 値]

GET は、データベースのレコードをベースキー、または索引を手掛かりに検索しアクティブ・エリア (ACTIVE) に出力するコマンド。

検索したレコードの情報は、"DBL" (コマンドのみ)、または "LIST" コマンドで画面に表示することができる。

DBLIST は、データベースよりレコードを検索すると共にそのレコードの項目名、タイトル、収録期間の一覧表を画面、またはハードコピーに出力する。

DBLIST の場合は、指定コードの上位概念によって検索が行われる。

また、下記の条件式を書けば自然語によるサーチを行うこともできる。

このコマンドを用いることによりデータベースの収録レコードの最新の情報が見られる。

- 条件式の定義を BNF 記法で下記に示す。

〈条件式〉 ::= 〈条件句〉 | 〈条件句〉 { 〈接続子〉 〈条件句〉 }

〈条件句〉 ::= 〈フィールド名〉 〈関係子〉 〈定数リスト〉

〈接続子〉 ::= & | ! | ^

但し、 & (論理積), ! (論理和), ^ (否定)

〈関係子〉 ::= = | ^ = | > | > = | < | < = | ~

但し、 ~ (区間指定子)

〈定数リスト〉 ::= 〈定数〉 | 〈定数リスト〉 〈定数〉

〈定数〉 ::= 〈数値定数〉 | 〈文字定数〉

〈数値定数〉 ::= 〈数字〉 | 〈符号〉 〈数字〉

〈符号〉 ::= + | -

〈文字定数〉 ::= 〈文字列〉 | 〈リテラル〉

〈文字列〉 ::= 〈英字〉 | { 〈英字又は数字〉 }

〈リテラル〉 ::= ' 〈文字列〉 '

- コード： データベースの系列コード。

頭2文字が "??" のカタログ名 (コード・リスト) も指定可。

- フィールド (欄) 名, 又は索引名:

"DISPLAY (又は D) I" コマンドで確認可。

- 関係子： =, ^=, >, >=, <, <=, 又は～（区間指定）。
- 値： 数字, 又は文字列（12文字以上はアボトロフィーで囲む）。
- TO（区間指定）, ADD（追加モード）, LP（プリンター）はキーワード。
- 関連コマンド： GETR（繰り返し形データ検索コマンド）。

〈GETコマンドの使用例〉

① GET CODE1, CODE2

2つのレコードを検索する。

② GET CODE1 TO CODE2

区間指定による検索。

③ GET C1 TO C2 NAME=GM,CHRYSLER

ベースキー（C1, C2）とレコードの内容による検索。

NAME は欄名で "D I" コマンドで確認できる。

〈DBL コマンドの使用例： IFS データベースの検索〉

① DBL

アクティブ・エリアのレコードの一覧表。

② DBL 158

日本（国コード158）の全項目の検索と表示。

③ DBL 1119 LP

米国の国民所得関係項目をハードコピーに出力。

>> PUTBASE（又は PB）, GETBASE（又は GB）

ベース・レコードの書込み（PB）と読取り（GB）

PB [名前, …… , 名前]

GB 名前, [, 名前, …… , 名前] [TO 名前] [ADD]

PB 文は, アクティブ・エリアのレコードをベース・エリアに登録する。

ベース・エリアにデータが出力されるのは PB コマンドの外 LET,COM文

の演算結果（左辺変数），LSM 系コマンドによって得られた推定値等の情報である。

一方 GB 文は，ベース・エリア（BASE）の指定レコードをアクティブ・エリアに呼び出す。

ベース・エリアのデータは，出力系，或は分析系コマンドで直接指定できる。

・名前： ベース・エリアの登録名。

LET 文，COM 文，U 文でも登録される。

PB 文で名前を指定しなかった場合は，レコード自身が持っている名前（データベースの系列コード等）で登録される。

・TO（区間指定），ADD（追加モード）はキーワード。

・関連コマンド：

UPDATE（又はU）；ベース・エリアのデータの作成または更新。

INFOBASE（又はIB）；登録諸元の一覧表。

① GB CODE1, CODE2 ADD

二つのレコードを追加モードでアクティブ・エリアに呼び出す。

② PB CODE3 CODE4

アクティブ・エリアの頭から2レコードをベース・エリアに登録する。

③ IB CDS TO CDT

CDS から CDT までのレコードの登録情報を表示。

>> GETCROSS（又はGETC）

検索した時系列データをクロスセクション・データに変換する

GETC コードA [コードA, ………, コードA]

REPEAT コードB [コードB, ………, コードB]

GETC ￥￥コード [ ￥￥コード, ………, ￥￥コード]

REPEAT ￥￥コード [ ￥￥コード, ………, ￥￥コード]

繰り返し形データ検索コマンドで、国・企業集団等を1組記述することで各集団の所要項目を一括して検索し、検索データをクロスセクション・データに変換しアクティブ・エリアに出力する。

GETCROSS 文の書き方は、GETREPEAT 文に準じる。

- コードA, コードB: コードA の # 部分にコードB の文字列が入れられる。
- 々々コード: コード・リスト (コード記述の集合) が登録されているカタログ名。

《GETCROSS 文に関する注意事項と使用例》

- 変換によって得られる横断面データは PERIOD 文で指定した最初の時点のデータである。
- 変換後のデータの起点表示が1に変えられる。

従って、再び時系列データとして用いる場合は、PERIOD 文の指定が必要である。

- これらの文を用いると表頭・表側情報 (コード名, 時点表示等) が不定となる。

〈使用例〉

① PERIOD 91 TO 93:

```
GETC #110000,#210000,#311000,#312000 REPEAT 9501,9502,
      9503:
```

3社4財務項目の行列 (1991年データ) に変換しアクティブ・エリアに出力。

《記》

文仕様と関連文の詳細は“TRANSPOSE 文”を参照せよ。

>> GETPUTC (又は GETP)

システム・カタログの取り込み

GETP [ユーザ・カタログ名 = ] システム・カタログ名 [LP]

システム・カタログの表示，またはシステム・カタログの内容をユーザ・カタログに取り込む。カタログされたこれらの名標は GET 系，及び DBLIST 文で用いることができる。

ユーザ・カタログの内容表示は INFO 文で，内容更新は UC（省略形）文で行うことができる。

- ・ユーザ・カタログ名：頭2文字が"¥¥"記号の名標。
- ・システム・カタログ名：頭3文字が"¥¥¥"記号の名標（読取りのみ）。

#### <使用例>

##### ① GETP ¥¥¥EXS

EXSTAT 関係の全システム・カタログの画面表示。

##### ② GETPUTC ¥¥EXSPL = ¥¥¥EXSPL

システム・カタログ名 "¥¥¥EXSPL" の内容をユーザ・カタログ名 "¥¥EXSPL" で登録する。

#### >> PUTUSER (又は PU), GETUSER (又は GU)

ユーザ・レコードの書込み (PU) と読み取り (GU)

PU 段落名 [START 収録開始年・期]

GU 段落名 [段落名, ……., 段落名] [ADD]

PU文は，アクティブ・エリアのレコードをユーザ・ファイル (USER) に登録する。登録開始時期は，START 句がなければ PERIOD 文，または ROW 文で設定して有ればそれが優先する。

条件に合致したレコードを登録する場合は LIST 文を用い SAVE 句に段落名を書く。

PU 文で登録した段落名は，MATRIX 文でも用いることができる。

また，ユーザファイルは MATRIX 文を用いても登録できる。

一方，GU 文は，ユーザ・ファイルに登録されている指定レコード群 (段落)

をアクティブ・エリアに転送する命令文である。

- 段落名： 第1文字が英字で7文字以内の英数字。
- 収録開始年・期： 収録開始時期を繰上げ、または繰り下げる場合に指定。
- ADD： 追加モードでアクティブ・エリアに転送される。
- 関連コマンド： MATRIX 文。

① PU UDATA

アクティブ・エリアの全レコードを UDATA の名前で登録する。

② GU UDT1, UDT2

段落 UDT1 および UDT2 をアクティブ・エリアに転送する。

③ IU UDS TO UDT

UDS から UDT までの段落の情報を表示。

>> GETREPEAT (又は GETR)

繰り返し形検索コマンド

GETR コードA [コードA, ………, コードA]

REPEAT コードB [コードB, ………, コードB]

GETR ￥￥コード [ ￥￥コード, ………, ￥￥コード]

REPEAT ￥￥コード [ ￥￥コード, ………, ￥￥コード]

繰り返し形データ検索コマンドで、国・企業集団等を1組記述することで各団体の所要項目を一括して検索しアクティブ・エリアに出力するコマンド。

- コードA, コードB：コードA の # 部分にコードB の文字列が入れられる。
- ￥￥コード：コード・リスト（コード記述の集合）が登録されているカタログ名。

UPDATECAT (または UC) コマンドで登録することができる。

〈使用例：CSSA データベースを用いた場合〉

① GETR #012 REPEAT GM000,IBM00

GM 社と IBM 社の売上高 (012) を検索する。

② GETR GM000#, IBM00# REPEAT 012

①と②は同じ項目が検索されるがレコードの順序は異なる。

③ GETR ￥￥C1 REPEAT ￥￥CB1

会社名と項目をカタログ名 ( ￥￥C1 及び ￥￥CB1 ) で指定する。

## >> HARDCOPY (又は HC)

実行結果のハードコピー出力

### HARDCOPY

当該コマンドを送信する直前に実行したコマンド集合を再度実行しその結果をハードコピーに出力する。ページ替を行う場合は、コマンド集合の中に適宜 PAGE コマンド、またはキーワードの PAGE を入れておく必要がある。

・関連コマンド: OUTH (ハードコピー・イメージを直ちに出力する)。

#### <使用例>

DBL 11165: PLOT: .....直前のコマンド集合。

① HC

直前のコマンドを再度実行しハードコピーに出力する。

## >> HELP (又は H)

使用の手助け

HELP [コマンド名 (省略形)] [ON] [OFF] .....①

HELP DB, データベース名 .....②

SECRETARY の使用の手助けをすると共に最も新しい情報を使用者に提供する。

・コマンド名: PD,G,L 等のように省略形で指定する。

コマンド名 (H) のみの場合は、命令文一覧表を表示する。

- ON : SECRETARY を説明付のモードとする。
- OFF : " 通常モードに戻す。
- データベース名 : 指定データベースの収録分野, 出所, 収録範囲, コード体系等に関する説明。

〈使用例〉

- ① H  
命令文と簡単な機能の一覧表の表示。
- ② H ON  
説明付きのモードにする。
- ③ H OFF  
通常モードに戻す。
- ④ H G  
GET コマンドの説明文を表示する。

>> HISTn (又は HISTnC)

ヒストグラムの表示

(a) 系列データを入力する場合

HIST1~3 [Bコード] [#(n)]  
[SUBMX (行列名/行リスト/列リスト/)] [LIST [SORT]]  
[WITHOUT ZERO] [LP] [PAGE]

(b) 横断面データを入力する場合

HIST6~8 [コード] [TO コード]  
[索引名 関係子 値, …… , 値] [接続子 条件句]  
[[FORM] (n) [V1] [V2]] [LIST [SORT]]  
[WITHOUT ZERO] [LP] [PAGE]

- コマンド : HIST1, HIST6 (1 $\sigma$ 間隔, +3 $\sigma$ )

HIST2, HIST7 (0.5 $\sigma$ 間隔,  $\pm 3\sigma$ )

HIST3, HIST8 (指定様式)

HISTnC (各コマンドの累積分布図)

- Bコード : ベース・エリアのコード (BASE)。
- # (n) : アクティブ・エリアの指定。n は位置指定の整数。
- 行・列リスト : 整数の並び (1~5の指定例 ; 1, -5)。
- LIST : 入力データの印刷。
- SORT : データを昇順に分類して印刷 (LIST の指定も必要)。
- WITHOUT ZERO : ゼロ, 又は -9999999のデータを入力から除く  
(又は ZERO指定のみでもよい)。
- FORM : ヒストグラムの出力様式指定。  
n (区間点の数), V1 (初期値), V2 (間隔の値)。  
n, V1, V2 は全てオプション指定 (但し位置パラメータ)。  
FORM のみの場合は, システムが標準値を設定する。
- コード : データベースのコード (桁数が少ない場合は上位概念により検索)
- 条件式 : [条件句] [接続子 条件句] …… [接続子 条件句]  
条件句 ; 索引名 (又は項目名) 関係子 値, …… , 値  
接続子 ; & (論理積) , ! (論理和), または, ^ (否定)。  
索引名 ; DISPLAYコマンドで確認できる (例 "D I")  
関係子 ; =, ^ =, >, >=, <, <=, または, ~ (区間指定)。  
値 ; 統計コード, 文字列 (12文字以上は, ' で囲む), 数値等。

#### 《記》

条件式の詳しい書き方は, GET 文の項を参照せよ。

#### <使用例>

① CFILE IFSQ: PD 86 TO 93: DBL 11163: HIST1 LIST:

国際資金データベース (IFSQ) より米国 (111) の卸売物価指数 (63) を検

索し、1986～1993年のデータに関して、 $1\sigma$ （標準偏差）間隔で分布図（HIST1）を描く。

HIST1 文でデータを指定していないので1番目のデータが選択される。

② CFILE KG1A: PD 91:

HIST7C 5400 TO 6499 INDEX1=50201 & DATA(91)～1000,5000  
ZERO:

企業財務データベース（KG1A）より、証券コードが5400～6499までの企業群で、1991年の従業員数（50201）が1000～5000人の会社について、従業員の分布図（HIST7）を $1\sigma$ 間隔で描く。

## >> 条件文

(a) IF v1 関係子 v2 THEN 文:

(b) IF v1 関係子 v2 THEN DO:

文1: 文2: …………… END:

論理条件（v1 関係子 v2）を評価しそれが真の場合に、文または文の並びが実行される。

(a) は単一の文の場合、(b) はブロック型で IF-DO 文から END 文までの文集合を制御する。

- v1: スカラー変数、または、ベクトル変数・#表記によるスカラー指定。
- v2: 定数、スカラー変数、ベクトル変数・#表記によるスカラー指定。
- 関係子: =, ^=, >, >=, <, または <=

## <使用例>

① IF @K=10 THEN PLOT #(1), #(2)

スカラー変数 "@K" が10ならば PLOT 文を実行する。

② DBL 111 TO 158 ITEM=99Z: LET R = 0:

DO @I=1 TO @COUNT:

```
IF #(@I/84)>100 THEN DO: DATA:
  LET R = (#(@I) - #(@I, -1)) / #(@I, -1):
  DATA #(@I), R: END: END:
```

IFS データベースより米国（国コード111）～日本（158）の各国の人口（99Z）を検索，1984年データが一億人（単位100万人）以上の国について人口増加率を計算し，人口と人口増加率を表示する。

@COUNT は，アクティブ・エリアの総レコード数が入れられるシステム定義のスカラー変数。

## >> INFO (又は I)

ユーザ情報表示コマンド

- (a) INFO (又は I) 登録名 [TO登録名] [LP]
- (b) INFOBASE (又は IB) 登録名 [TO 登録名]
- (c) INFOUSER (又は IU) 登録名 [, 登録名, ...]  
[TO 登録名] [ALL]

ユーザファイルに登録のカタログ，段落，行列，またはベース・エリアに登録されている変数に関する情報を表示する。

- (a) は命令文集合，コードリスト等のカタログ情報。
- (b) はベース・エリアに登録の変数に関する情報の一覧表。
- (c) は，ユーザ・ファイルに登録の段落，行列に関する情報の一覧表。

ALL: 段落，または行列の個々のレコードに関する属性も表示。

- 登録名: 行列，または段落名。
- ALL : 行列，または段落の個々のレコードに関する属性も表示。
- TO, LP, ALL は，キーワード。

LP はハードコピーへ出力する場合，TO は区間で表示する場合に用いる。

### <使用例>

① I ¥ TO Z

登録の全カタログ（コマンド、コードリスト等）の表示。

（名前 "¥" で登録したレコードが有るとする）

② IB ¥ TO Z

ベース・エリアへの登録状況を表示。

③ IU ¥ TO Z

ユーザ・ファイルへの登録状況を表示。

④ IU MX1 TO MX5, VECT

MX1~MX5 および VECT の登録情報を表示。

>> LET, COMPUTE (又は COM) : 配列演算

LET 識別子 = 代数演算式

配列演算または要素（スカラー）演算を行う命令文。

右辺には、FORTRAN 言語同様にカッコ、組込関数を用いた演算式を書くことができる。FORTRAN の演算は要素単位に行われるが、このコマンドは配列を単位として行われる。

既に定義済みの変数名を左辺に書き COMPUTE 文で演算を行うとそのレコード自身が持っているタイトル情報を残す。LET文を用いた場合はタイトルはクリアされる。

- ・ 識別子: ベースエリアへの登録名（変数名）。

または、アクティブ・エリアを指定する "#"（又は "#Y"）表示。

《識別子の書き方の例（配列）》

- ① 変数名; BN(,,-1).....1期ラグ。
- ② # 表示 ; # (2).....アクティブ・エリアの2行目データ。  
# (3/83,91).....3行目の83年~91年データ。
- ③ # Y 表示; # Y(88).....88列（年）のデータ。

## 《識別子の書き方の例（要素）》

① 変数名; BN(5) ……………5番目の要素。

② # 名; # (4/89) ……………4行目, 89列 (年) の要素。

• 上の識別子の外, スカラー変数 (第1文字が @), 定数が書ける。

• 組込関数;

ABS (絶対値), COR (相関係数), COS (余弦), COV (共分散), CUMU (累積), DEV (偏差積和), EPA (季節調整: 経済企画庁版), EXP (指数), EXPECT (期待値), LOG (自然対数), LOG10 (常用対数), MEAN (平均), NOWB (ベース・エリアデータ数), NOWD (カタログ語数), POWM (データ系列・累乗), PROD (積和), SIN (正弦), SQRT (平方根), SDEV (標準偏差), TAN (正接), TBLN (正規分布確率), TOTAL (合計)

## &lt;使用例&gt;

① LET R = (DT-DT(,,-1)) / DT(,,-1)

項目名 "DT" の対前年伸び率の計算。

② COM ANS = EPA(#(2)+ #(4))

アクティブ・エリアの指定と関数 (EPA: 季節調整) の使用。

## &gt;&gt; LINE

コマンドの表示番号を変更する

## LINE n

コマンドの表示番号を, nからにする。

プロシージャ実行の都度コマンドの行番号が n から始まる。

## &gt;&gt; LIST (又はL)

レコード属性の表示, ユーザファイルへの登録。

LIST [条件式] [SAVE {段落名}] [LP]  
 [ACTIVE]

GET, DBL 文等で検索されアクティブ・エリアに入れているレコードを画面またはハードコピーへ出力する。条件式を書けば条件に合致したレコードのみ出力する。

また、SAVE 句の指定で条件に合致したレコードのみ登録（ユーザファイル）することもできる。

・条件式：GET・DBLIST コマンドの項を参照せよ。

① LIST TEXT = GOLD

TEXT 欄が "GOLD" レコードを画面に表示する。

② L TEXT = GOLD SAVE UNAME1

ユーザファイルへも "UNAME1" の段落名で登録する。

③ L DATA(92)~23,45

92年のデータの値が23~45の範囲にあるレコードを表示する。

## >> LISTOPTION (又はLOP)

コマンドの表示 (印刷) 制御

LOP abcd

警告メッセージの抑制 (1 又は 0), コマンドの表示・印刷抑制 (1 又は 0) 等 SECRETARY 実行過程における操作上のオプションを処理系に与える。

### 《書き方の注意》

桁の上位を指定する場合は、下位の指定も必要である。

例えば c を指定する場合は、d も併せて指定しなければならない。

### 〈使用例〉

- (1) LOP 10: コマンドの表示・印刷を抑制し、警告メッセージは表示する。
- (2) LOP 1: 警告メッセージを抑制する。

(3) LOP 11: コマンドの表示・印刷, 及び, 警告メッセージも抑制する。

## >> 最小自乗推定関係 コマンド

コマンド [識別子, ……] [TO 識別子] [PAGE] [LP]

コマンド 小行列関数 [PAGE] [LP]

最小自乗推定法によって線形回帰モデルの母数を求めると共に, モデルの適合性検定のための各種の計数値を出力する。

### • コマンド

- ① LSM, 又は LSMS (プロット出力無)。
- ② LSMN, 又は LSMNS (プロット出力無) ……非定数項モデル。
- ③ LSMC, 又は LSMCS (プロット出力無) ……定数項付モデル。

### • 識別子 (最初に指定した変数が被説明変数) :

- ① ベース・エリアの変数名。
- ② # (i) ; アクティブ・エリアの i 行目のレコードの指定。
- ③ #Y (j) ; アクティブ・エリアの j 列目のレコード。

### • TO : 識別子を区間で指定する場合に用いる。

### • 推定値 (外挿を含む), 残差 (外挿分を含む), t 値はそれぞれ ¥EST, ¥RESI, ¥TVAL の名前でベース・エリアに出力される。

(LSM 及び LSMS コマンドを除く)

外挿期間は, 被説明変数データの充足している期間に渡って行われる。

推定値等の値は, PERIOD (または ROW) 文を所定の期間に設定した後, データ出力形コマンドで表示させることができる。

但し, ¥TVAL への t の値の出力は, 配列の 1 番目からである。

## <使用例>

### ① LSMCS SYOHI SYOTOK

SYOTOK を説明変数, SYOHI を被説明変数とし回帰分析を行う。

② LSMN YY # (3) TO # (6) LP PAGE

YY を被説明変数，アクティブ・エリアの3番目～6番目を説明変数として  
回帰分析を行い，結果をハードコピーに改ページして出力する。

《記》

プログラムの出所は：

(1) LSM, LSMS

「HSAP統計計算ライブラリー」，HITACマニュアル (800-7-009) 重回帰。

(2) LSMN, LSMNS, LSMC, LSMCS

「数値計算副プログラムライブラリー」，HITACマニュアル (8080-7-141)，  
MSL II，第3分冊-統計計算，回帰分析を基に作成。

従って，解法・出力される情報の定義，エラーメッセージ等に関しては，  
それぞれのマニュアルを参照せよ。

>> MATRIX (又は MAT)

行列演算文

**MATRIX 行列 (変数) 名 = 行列演算式**

行列演算 (マトリックス)，配列演算 (ベクトル)，要素演算 (スカラー) を  
行う命令文。右辺には組込関数を用いることができ行列表記法に沿った式を書  
くことができる。

また，小行列関数により行列の結合 (行又は列) も可能である。

・行列名：第1文字が英字で7文字以内の文字列。

行列が既に登録されている場合には，当該行列演算式で定義される値に更  
新される。

尚，転置行列は行列名にアポストロフィ (') を付け，逆行列は行列名に  
ダブルアポストロフィ (") を付けることで表される。

例；MX'，MX(1,3/3,6)'，MX"。

- ・行列演算式：識別子としては、行列名、スカラー変数（第1文字が @）定数、カッコ、関数名、および、演算子を書くことができる。

### <使用例>

① MAT XVECT = MX1" \* YVECT

連立一次方程式の解。

② MAT BETA = (X' \* X)" \* X' \* Y

$y = X\hat{\beta} + e$  の回帰係数 ( $\hat{\beta}$ ) を算出。

③ MAT MX = SQRT(X)

行列 X の全要素の平方根を行列 MX に入れる。

④ MAT CORMX = CORX(X)

行列 X の相関行列を求め CORMX に入れる。

⑤ MAT MX = (IDENT(8)-A)"

A 行列 (8×8) のレオンチェフ行列を求める。

⑥ MAT M(2,4/2,3) = X(4,5/1,3)'

X の小行列の転置行列を行列 M に詰め込む。

⑦ MAT M = SUBMX(X/1,3,-7,9/1,3/)

行列 X の1・3～7・9行, 1・3列を行列名 M で登録する。

### >> MODE

モードの設定

MODE [MENU メニュー・ファイル名] [ACTIVE]

動作モードの設定、或は切換えを行う命令文で随時利用できる。

メニューは、SECRETARY 言語で記述されたものを予め TSS ファイルに登録しておかなければならない。メニューの内容は、"D MENU" でみられる。

メニュー・モードでは：

- ① メニューの番号選択。
- ② メニューの内容の表示（必要ならば画面で修正する）。
- ③ 実行（送信）のパターンで分析をすすめる。

- ・メニュー・ファイル名： 区分編成方式データセットのメンバー名。
- ・ACTIVE： 通常モード（省略時選択）。

〈使用例〉

① MODE MENU EX1

EX1 に登録のコマンドに従ったモードにする。

② MODE

メニュー・モードを通常モードに戻す。

>> MOVINGAV (又はMAV)

MAV [識別子 = 識別子] [SPAN p]

移動平均法によってデータの平滑化を行う。コマンド (MAV) のみの場合は、アクティブ・エリアの全レコードの移動平均がとられ元のレコード位置に入れられる。

- ・識別子： ベース・エリアへの登録名，または，#（または#Y）表示。
- ・p（インターバル）： 平均値算出の区間。省略すると PERIOD 文の SPAN 句の設定情報が参照される（"D PD" で確認できる）。

〈使用例〉

① MAV MVDT1 = DT1

DT1 の移動平均をとり MVDT1 に入れる。

>> MXLIST (又は ML)

行列データの出力（その1）

MXLIST (又はML) [行列名] [行開始点, 列開始点]

行列データを画面，またはハードコピーに出力する。

行列サイズが大きい時は，ファンクション・キーを用いてスクロールをして入力する。

- 行列名： 第1文字が英字で7文字以内の文字列。  
小行列表記及び転置指定（例；MX'）を行うことができる。  
省略するとアクティブ・エリアのテーブルが指定される。
- 行開始点，列開始点： 最初に表示する位置（デフォルトは1行・1列）
- ファンクション・キー（日立T-560/20，HT-5455等の端末の場合）：  
F7（左移動） F8（右移動） F9（上移動），F10（下移動）。  
F6（画面表示の列をハードコピーに出力する）。

#### <使用例>

##### ① MXLIST MX

行列 MX を表示する。

##### ② ML MX(11,25/5,10)

行列 MX の部分行列（11～25行，5～10列）を表示する。

#### >> MXUPDATE（又は MU）

行列データの作成・更新

##### MXUPDATE 〔行列名〕 〔行開始点，列開始点〕

行列のタイトル，行・列サイズ，表頭，表側，行列データ等を2つの画面で作成ないし更新するコマンド。前三者を第1画面で，表頭以下を第2画面によりカーソルを移動して作成ないし変更する。第1画面から第2画面への移行は送信（ENTER）キーを用いる。

行列サイズが大きい時は，スクロール（ファンクション・キー）操作によりデータの更新を行うこともできる。

また，当該コマンドは既作成の行列でもサイズの拡張・縮小が可能である。

- 行列名： 第一文字が英字で7文字以内の文字列。  
省略するとアクティブ・エリアのテーブルが指定される。
- 行開始点 列開始点： 最初に表示する位置（デフォルトは1行1列）。
- ファンクション・キー（日立T-560/20, HT-5455等の端末の場合）：  
F7（左移動），F8（右移動），F9（上移動），F10（下移動）。  
F6（画面表示の列をハードコピーに出力する）。
- コマンドの取消しは割り込みキーを用いる。

>> NE

無効コマンド

機能部にこのコマンドが書かれたステートメントは何の働きもしない。ステートメントの頭に同コマンドを入れることによりステートメントを無能にする。通常は、プログラムテスト時に用いられる。

>> NEWCON (又はNEWCOFF)

新世紀（西暦2000年への対応を行う

§ § 新世紀に当たってユーザの行う手続き

◎ 設定コマンド: NEWCON (解除コマンド: NEWCOFF)

- 設定以後期間設定文 (PERIOD) は、西暦年4桁で指定する。

≪例: PERIOD 1955 TO 2000

PD 1996.01 TO 2000.02

- NEWCON コマンド設定有無は、"D SYST" で確認できる。

§ § NEWCON コマンド設定によって "SECRETARY" の行う処理等

- 1) ベース・エリアの全レコードのタイムスタンプを西暦年4桁にする。  
"IB" コマンドで確認できる。
- 2) ユーザ・ファイル（段落）は新世紀対応未処理のため再登録が必要。  
"IU" コマンドで確認できる。

### <使用例>

#### ① NEWCON

新世紀への対応を行う。

#### ② NEWCOFF

新世紀への対応を解除する。

### >> NOPAGE

画面制御の解除

#### NOPAGE

画面制御（ページ替え）を禁止する。回復はできないので画面制御を再度行わせたい場合は、SECRETARY を終了し再起動する。

端末エミュレータがラインエディタの場合は、SECRETARY 起動後このコマンドを最初に入れておく必要がある。

### >> OPEN

ファイルのオープン文

#### OPEN ファイル名

GETR や GET 文で検索されたレコードは、TMPF ファイルにも出力される。このファイルをオープンするコマンドが OPEN 文で必要に応じて何回でも用いることができる。

TMPF は順処理用に定義されたファイルで、DO・READTMPF・READSUM 文等を用いて逐次（バッチ）的処理を行うことができる。

- ・ファイル名： TMPF, TMPF3。
- ・関連文： DO 文; DO WHILE(FLAG(1)=0) の形で用いる。  
RDT (又は READTMPF) 文; TMPF ファイルの逐次読み取り。

RDS (又は READSUM) 文; 累積形の逐次読み取り。

<使用例>

① OPEN TMPF:

TMPF ファイルを開く。

《例: 各企業の財務比率を繰り返し算出しプロットをする》

(CSSA ファイルを用いた場合)

GETR #060, #130, #006 REPEAT GM000, F0000, IBM00:

OPEN TMPF:

DO WHILE(FLAG(1)=0):

RDT: LET R = #(1)+ #(2)+ #(3):

PLOT R: END:

>> OUTDEV: 出力装置の定義

OUTDEV [LP] [LPRINT]

データの出力, 分析結果, 或はユーザ情報・システム情報等を出力する装置を定義する。

LP または LPRINT を指定して出力した情報は, SECRETARY のセッションを終わった後 (/\* 送信後) ハードコピーに出力される。

コマンドのみの場合は出力を画面表示に戻す。

出力系コマンド (DATA,DTM,LSM等) にオプションとして LP を書くこともできる。その場合はそのコマンドの実行に関してのみハードコピーへ出力される。

- LP: 出力装置をハードコピーとする (コマンドは画面にも表示する)。
- LPRINT: " (コマンドを画面に表示しない)。
- 関連コマンド: OUTHC。

<使用例>

## ① OUTDEV LP

出力装置をハードコピーとする。

## ② OUTDEV

出力を画面表示に戻す。

## &gt;&gt; OUTHC (又は OUTH)

実行結果を直ちに高速プリンタへ出力

## OUTHC

"OUTDEV LP", または, 出力系コマンドで "LP" を指定して出力したハードコピー・イメージを直ちに高速プリンタに印書する。

当該コマンドは, SECRETARY のセッションで必要に応じて何回でも用いることができる。

《付記: 本経済経営研究所の場合の出力クラス》

SECRETARY の呼び出し手続きで指定できる出力クラスは, A・H (連続紙) K・L (カット紙) でそれ以外は L クラスとして扱う。

尚, "H" および "L" クラスは, ホールド・ジョブである。

## &lt;使用例&gt;

## ① OUTHC

直ちに高速プリンタに印書する。

## &gt;&gt; OVER (又は /\*)

"SECRETARY" セッションの終了

## OVER [TRANS]

## /\* [TRANS]

SECRETARY のセッションを終わらせ, TSS コマンド・モードに戻す。

## &lt;使用例&gt;

① OVER TRANS

SECRETARY 終了時にアクティブ・エリアのデータを反転出力する。

>> PAGE (又は PG)

改ページ

PAGE

ハードコピーの用紙を改ページする。

>> PERIOD (又は PD)

分析期間の定義

(a) PERIOD [分析開始期] [TO 分析終了期] [BY 期種] [SPAN i]

(b) ROW [開始行] [TO 終了行]

(a) 分析期間, 期区分 (年次, 半期, 四半期, 月次) の定義, 或いは, MOVINGAV (移動平均), AGS・AGM・AGK (アグリゲート) コマンド等が参照する情報 (SPAN i) を定義する。

次の PERIOD 文までこの情報が保たれる。

(b) 行の範囲の定義, PUTUSER (又は PU) 文を用いて段落や行列の登録を行う場合に行の登録区間の情報を与える。

- 分析開始・終了期: 西暦年で与える, 年次データ以外は, ××.××の形で与える。
- 期種: A (年次又は項目), H (半期), Q (四半期), M (月次)
- i : 自然数 (間隔尺度を与える)。
- 開始行・終了行: データの収録開始点を1としてオフセット・ポイント値を自然数で与える。

〈使用例〉

① PD 81 TO 90 BY A

1981年から1990年までの年次データを分析期間とする。

② PD TO 93.12

終了年・期を、1993年12月とする。

③ PD @START TO @END

スカラー変数で開始・終了時点を与える。

④ PD SPAN 4

間隔尺度を4とする。

⑤ ROW 1 TO 15

1行から15行までとして定義する。

## >> PLOT (又は PL)

データ出力系コマンド (プロット出力)

(a) PLOT [識別子, ..., 識別子] [TO 識別子] [識別子, ...]

[DECIMAL p] [PAGE] [LP] [SKIP n]

(b) PLOT 小行列関数 [DECIMAL p] [PAGE] [LP] [SKIP n]

(a) は、ベースエリア、またはアクティブ・エリアのデータを縦型の出力様式で画面、或はハードコピーに出力する。

(b) は、行列 (又は段落) のデータを出力する時に用いる。

この場合データの収録開始点を1と見なして小行列関数に指標を与える。

- ・ 識別子： ① ベース・エリアの変数名。
- ② #(i): アクティブ・エリアの i 行目のレコードの指定。
- ③ #Y(j): アクティブ・エリアの j 列目のレコード。
- ・ 小行列関数： 小行列関数 (SUBMX) を参照せよ。
- ・ p: 出力データ表示の小数点以下桁数 (自然数)。
- ・ PAGE (改ページ), LP (ハードコピー出力), SKIP (改行)。
- ・ 関連コマンド (括弧内は省略形)：

DATA(DT), DATAG(DTG), DATAMPL(DTM),  
SCATTER(SC), SCATTERN(SCN)

関連コマンド全体については“データ出力系コマンド(その1)”を参照せよ。

〈使用例〉

① PLOT

アクティブ・エリアの全レコードのプロット表示。

② PL B1 # (3) # (5)

アクティブ・エリアとベース・エリアのレコードの混在出力。

>> PUTCATALOG (又は PC)

アクティブ・エリアのレコードの指定情報を編集出力

PC カタログ名 = 欄名 [FORMAT = '編集書式']

[WITH 乱数形式, 自然数]

アクティブ・エリアの指定情報を編集してカタログファイルに出力する。

この機能によってデータベースやユーザファイルに収録されているコード、テキスト等の情報を SECRETARY のコマンドで用いることができる。

- カタログ名： 選択・編集した情報を登録する名前。
- 欄名 (フィールド) 名： 情報を抽出する欄 (“D I” コマンドで確認可)。
- 編集書式： ” 記号で囲んだ文字列は、そのまま出力する。  
An 編集子; 指定欄のn文字を出力する。  
Xn 編集子; 指定欄のn文字を読み飛ばす。
- 乱数形式： RNU (一様乱数), RNN (正規乱数),  
REN (指数乱数), RNP (ポアソン乱数)。
- 自然数： 発生乱数に対応して抽出するアクティブ・エリアのレコード総数。

① PUTCATALOG ¥¥PCA = ITEM

ITEM 欄の情報を編集なしで ¥PCA の名前で出力する。

② PC ¥P = BKEY FORMAT = 'A3, "#, "'

BKEY 欄の情報の頭3文字に文字列 (" #") を後に付けて出力

>> PUTBASE (又は PB)

GETBASE (又は GB)

ベース・レコードの書込み (PB) と読取り (GB)

《記》コマンドの仕様は, "GETBASE" を参照せよ。

>> PUTCROSS (又は PUTC)

アクティブ・エリアのデータの最初の時点のデータを1ケースとして出力

PUTCROSS(PUTC)

アクティブ・エリアのデータの内 PERIOD 文で指定した最初の時点のデータを1ケースとして外部ファイル (SCWORK1) に出力する。

《PUTCROSS 文に関する注意事項と使用例》

- この文を用いると表頭・表側情報 (コード名, 時点表示等) が不定となる。
- 変換によって得られる横断面データは PERIOD 文で指定した最初の時点のデータである。
- PUTC 文は, アクティブ・エリアの初年度データを1レコードに編集し SECRETARY の "SCWORK1" ファイルに出力する。  
SECRETARY セッション中一度以上 PUTC 文を用いると当該文による出力が優先し終了時に "SCWORK1" ファイルに出力される。
- PUTC文による出力レコードを一時的に貯えるファイルとして "TMPF3" ファイルを用いる。
- 通常は, 国際経済経営データベースのデータを SPSS や SAS に引き渡す

時でケース数が多い場合に用いる。

〈使用例〉

① PERIOD 91 TO 93:

```
GETR # 110000, # 120000, # 210000, # 311000, # 312000 REPEAT 9501,
9502,9503:
```

```
OPEN TMPF; TMPF3:
```

```
DO WHILE(FLAG(1)=0):
```

```
RDT REPEAT 5:
```

```
DO @I=1 TO 5: LET #(@I+5) = #(@I)/10: END:
```

```
PUTCROSS: END:
```

興銀財務データベースより3社5財務項目を検索し、1991年データに関して各会社5項目の原データと、各項目の 1/10 の値の5項目を追加した10項目より成るレコード（1ケース）を "TMPF3" ファイルに出力する。

"TMPF3" ファイルに出力されたレコードがあると、SECRETARY 終了時に編集して "SCWORK1" ファイル（アクティブ・エリア）に出力する。

《記》

文仕様と関連文の詳細は "TRANSPOSE文" を参照せよ。

>> PUTTABLE (又は PUTT)

検索したデータを"スプレッド表"に編集出力する。

>> PUTUSER (又は PU)

GETUSER (又は GU)

ユーザ・レコードの書込み (PU) と読み取り (GU)

《記》コマンドの仕様は、"GETUSER" を参照せよ。

>> RANDUNIF (又は RNU) : 一様乱数

RANDNOR (又は RNN) : 正規乱数

RANDEXP (又は RNE) : 指数乱数

RANDPOIS (又は RNP) : ポアソン乱数

(a) RNU 下限値 TO 上限値 [INIT 初期値]

RNN 平均 標準偏差 [初期値]

RNE 平均 [初期値]

RNP 平均 [初期値]

(b) コマンド スカラー変数

一様乱数の仕様の定義及び生成 (a), 或いは1個の乱数を生成する (b)。

前者のコマンドの実行では乱数定義と共にベースエリアの ¥RANDOM に120~360個の乱数列を出力する。

- 下限値・上限値: 生成乱数の区間を指定する。
- 初期値: INIT 句を省略すると初期値として584287を仮定する。  
(桁数の大きい素数を用いると特性のよい乱数が得られる)。
- スカラー変数: 生成乱数を出力する変数名。
- 関連コマンド: INFO(I), PUTCATALOG(PC)。

<使用例>

① RNU 1985 TO 2000

1985~2000までの数値の一様乱数の仕様を定義すると共に ¥RANDOM に120個以上(その時のファイル定義によって変わる)の乱数列を出力する。

② PD 1 TO 10: DATA ¥RANDOM:

①で生成した乱数を10個表示する。

③ RNU @R

乱数1個を生成しスカラー変数"AR"に入れる。

>> READMT (又はRDM)

磁気テープデータの入力

READMT

磁気テープの SECRETARY フォーマットのレコードをアクティブ・エリアに入れる。

>> READTMPF(RDT), READSUM(RDS)

“TMPF” ファイルの逐次読み取り

READTMPF [REPEAT n]

READSUM [REPEAT n]

GETR や GET 文で検索されたレコードは、TMPF ファイルにも出力される。

このファイルを逐次的にnレコードずつ読んでアクティブ・エリアに出力するコマンドが READTMPF である。

READSUM 文は、既にアクティブ・エリアに入れられている n レコードに、新たに当該コマンドで読まれた対応するnレコードそれぞれが累積される。

(ACTIVE(i)+TMPF(i)→ACTIVE(i))。

- n: 1 コマンドで読み取るレコード数、省略した場合は、GETR 文のコードA部 (CETR 文の説明参照) のレコード数とみなされる。
- 関連文: OPEN TMPF 文; TMPF ファイルをオープンする。

>> RENAME (又は REN)

カタログ名の変更

REN カタログ名 (旧) カタログ名 (新)

カタログの名称を変更する。

- カタログ名: 第一文字は英字で12文字以内の英数字。

- 関連コマンド： INFO, SAVE, UC, DEL。

### <使用例>

#### ① REN PROC1 PROC2

カタログ名 "PROC1" を "PROC2" に変更する。

### >> SAVE (又は S)

命令文集合の登録

#### SAVE プロシージャ名

直前に SECRETARY の処理系に送信した命令文集合を指定したプロシージャ

(カタログ) 名で登録する。

指定したプロシージャが既に登録されている場合は、内容が更新される。

- プロシージャ名： 第1文字は英字で12文字以内の英数字。
- 関連コマンド： INFO; カタログの一覧表。

UC; コマンド・プロシージャを更新する。

DEL; コマンド・プロシージャを削除する。

REN; コマンド・プロシージャの名称を変える。

### <使用例>

#### ① SAVE PROC1

直前の命令文集合を, "PROC1" の名前でカタログする。

### >> SCATTER, SCATTERN (又は SC, SCN)

データ出力系コマンド (複数系列・縦出力)

(a) SCATTER [識別子, ..., 識別子] [TO 識別子] [識別子, ...]

[DECIMAL p] [PAGE] [LP] [SKIP n]

(b) SCATTER 小行列関数

[DECIMAL p] [PAGE] [LP] [SKIP n]

(a) は、ベースエリア、またはアクティブ・エリアのデータを散布図の様式で画面、或はハードコピーに出力する。

(b) は、行列（又は段落）のデータを出力する時に用いる。

この場合データの収録開始点を1と見なして小行列関数に指標を与える。

• SCATTER（又は SCN）は、基礎変数（1番目の変数）による縦軸のスケールは行われないうで、基礎変数のデータの並びで縦軸が割当られる。

• 散布図のプロット記号

クロス変数の番号とプロット記号との関係を次に示す。

クロス変数番号： 1 2 3 4 5 6 7 8 9 10

プロット記号： \* 2 3 4 5 6 7 8 9 A

プロットが重複する場合は、“ロ”記号

• 識別子： ① ベース・エリアの変数名。

② #(i)： アクティブ・エリアの i 行目のレコードの指定。

③ #Y(j)： アクティブ・エリアの j 列目のレコード。

• 小行列関数： 小行列関数 (SUBMX) を参照せよ。

• LP (ハードコピー出力)。

• 関連コマンド (括弧内は省略形)：

DATA(DT), DATAG(DTG), DATAMPL(DTM),

PLOT(PL)

関連コマンド全体については“C5.2 データ出力系コマンド (その1)”を参照せよ。

〈使用例〉

① SCATTER B1 TO B5

ベース・エリアの B1 を基礎 (基底) 変数とし B2 から B5 までのレコードをクロス変数とする散布図を出力する。

## ② SCN

アクティブ・エリアの1番目のレコードを基礎変数とし、2番日以降のレコードをクロス変数とする散布図を出力する。

縦軸のスケールリングは行われず基礎変数のデータの並びで縦軸が割られる。

## &gt;&gt; SCATTERS (SCS),SCATTERX(SCX)

## 構造図の作成

SCATTERS (又は SCS) 小行列関数 …………… (a)

SCATTERX(又は SCX) 小行列関数 …………… (b)

[VALUE 最小値, 最大値] [LP] [PAGE]

行列データを散布図の形で画面、またはハードコピーに出力する。

多変量解析の構造図出力等に使用される。

(a) 基礎変数 (第1列ベクトル) に対する複数のクロス変数 (第2列～n列ベクトル) との関係を構造図で表す。

クロス変数の番号とプロット記号との関係を次に示す。

クロス変数番号 : 1 2 3 4 5 ……………

プロット記号 : 1 2 3 4 5 ……………

プロットが重複する場合は, “ロ” 記号

(b) 基礎変数 (第1列ベクトル) に対するクロス変数 (第2列ベクトル) との関係を構造図で表す。

ケース (標本) の番号とプロット記号との関係を次に示す。

ケース (標本) : 1 2 3 …… 9 10 11 12 …… 35 36 37 38 …… 61

プロット記号 : 1 2 3 …… 9 A B C …… Z a b c …… z

プロットが重複する場合は, “ロ” 記号

《条件・制限》 ケース数  $\leq$  61

- 小行列関数：小行列関数の表記法による（4.2.7 (2) を参照せよ）。
- 最小値，最大値：軸のスケールリング。

〈使用例〉

① SCS SUBMX(X/2,3/1,-9/)

X行列の2・3行，1～9列の表示。

② SCS SUBMX(X) VALUE -3,5

X行列（全変数）を最小値 -3，最大値 5のスケールで表示する。

>> SKIP

改行

SKIP n

ハードコピーへの出力で，行間に n 行の空白を入れる。

>> SORT

アクティブ・エリアのレコードの分類

SORT 欄名 [, 欄名, …… , 欄名] [BY D]

GET, GETREPEAT, GETBASE 等のコマンドで出力されたアクティブ・エリアのレコードを指定欄で分類（昇順）する。データの値による分類もできる。

- 欄名：”DISPLAY (又は D) I” コマンドで確認できる。

欄名の代わりに ”P,I” の組としても指定可能である。

P (オフセットポイント) , I (欄の長さ)。

- BY D: 降順で分類する。

〈使用例〉

① SORT TEXT

”TEXT”欄で昇順に分類する場合に指定する。

② SORT 1,12

オフセットポイント、欄の長さで指定した場合。

### ③ SORT DATA (93) BY D

1993年のデータ値により降順に分類する。

## >> データ出力系コマンド (その1. 1)

STATIS (又は ST)    [識別子, ………, 識別子] [TO 識別子]  
                          [DECIMAL p] [PAGE] [LP]

ベース・エリア, またはアクティブ・エリアのデータ系列の最小値・最大値・平均値・標準偏差を画面, 或はハードコピーに出力するとともに, それぞれ ¥MIN・¥MAX・¥MEAN・¥STD・の名前でベース・エリアにレコード数に対応する並び (横断面データ) で出力する。

出力された系列の起点は1に定義されるので, それらのデータを用いるには期間設定 (PERIOD文) をし直さなければならない。

¥MIN, ¥MAX, ¥MEAN, ¥STDを時系列として出力する場合は, "ST #Y" とする

(アクティブ・エリアのデータの場合)。

・識別子:

- ① ベース・エリアの変数名。
- ② #(i); アクティブ・エリアの i 行目のレコードの指定。
- ③ #Y(j); アクティブ・エリアの j 行目 (時系列データならば時点)

全期間を指定する場合は, "ST #Y" とする。

- ・p: 出力データ表示の小数点以下桁数 (自然数)。
- ・PAGE (改ページ), LP (ハードコピー出力)。

### <使用例>

#### ① STATIS

アクティブ・エリアの全レコードの簡易統計表を出力する。

## >> SUMMATION (又は SUM)

アクティブ・エリアのレコードの集計

### SUMMATION 識別子 [ = #Y ]

アクティブ・エリアの総計 (縦集計) を算出する命令文。

" = #Y" を指定すると表の横集計を行う。

- 識別子: ベース・エリアへの登録名, または, # (i)・#Y(j) 表示。

### <使用例>

#### ① SUM DTOTAL

アクティブ・エリアのレコードの総計系列を "DTOTAL" に入れる。

## >> TITLE

タイトル印刷とハードコピー制御関係

### (a) TITEL (又はT) "タイトル"

### (b) TITEL (又はT) メンバー名 [開始行, 終了行]

出力帳票へ指定タイトル等を挿入する。

- メンバー名: システム・データ用データセットのサブファイル名。  
開始行, 終了行でサブファイル内の位置を指定する。

## >> TRANSPOSE (又は TR)

データの転置, および, その関連文。

### (a) TRANSPOSE [REPEAT n] [SAVE 行列名]

### (b) GETCROSS(GETC) 文

### (c) PUTCROSS(PUTC) 文

### (d) OVER TRANS

- n: 1行 (サンプル) の中に入れる列数 (項目数)。
- 行列名: 転置した行列を "USER" ファイルに保存する場合に指定。

データベースより検索した時系列データをクロスセクション・データに変換する文で次の4つの場合に分けられる。

(a) は、GETREPEAT (又は GETR) 文でレコードを検索した直後に TRANSPOSE文を用いてクロスセクション・データに変換する。

(b) は、GETREPEAT 文の代わりに GETCROSS (又は GETC) 文を用いることでクロスセクション・データに変換される。

GETCROSS 文の書き方は、GETREPEAT 文に準じる。

(c) は、アクティブ・エリアのデータの内 PERIOD 文で指定した最初の時点のデータを1ケースとして外部ファイル (SCWORK1) に出力する。

(d) は、SECRETARY のセッション終了時機能で "TRANS" を指定して終わることでアクティブ・エリアのデータが反転出力される。

(a) ・ (b) ・ (c) の場合、変換によって得られる横断面データは PERIOD 文で指定した最初の時点のデータである。

尚、(a) ・ (b) の場合は変換後のデータの起点表示が1に変えられるので再び時系列データとして用いる場合は、PERIOD 文の指定が必要である。

#### 《TRANSPOSE・GETCROSS 文に関する注意事項と使用例》

(1) これらの文を用いると表頭・表側情報 (コード名, 時点表示等) が不定となる。

(2) GET 文等でデータを検索し、TR 文で REPEAT 句を指定してデータを反転することはできるが、DMX 文でそのデータを表示させた場合、表頭・表側は意味の無い情報を表示する。

但し、データに関しては正しい横断面データを表示し、また、SAVE 句での登録も正しく行われる。

(3) GETR 文で検索できないレコードがある場合に、TR コマンドを用いても正しい横断面データはつくられない。

(4) DMX 文で表頭・表側を使用者の指定する名前に表示させたい時は

MXUPDATE (又は MU) 文でその情報を行列 (或は段落) に入れることができる。登録した行列を削除しない限り行列のデータ部分を変えても表頭・表側情報は変わらない。

〈使用例〉

① PERIOD 81 TO 84:

GETR #110000, #210000, #311000, #312000 REPEAT 9501, 9502, 9503:  
TRANSPOSE:

3社4財務項目の行列 (1981年データ) に変換しアクティブに出力。

GETR を GETC とすれば TRANSPOSE 文を省略できる。

② GET 9501 TO 9509 INDEX1=110000, 211000, 212000, 312000:

TR REPEAT 9:

索引を用いて検索した項目・会社順 (4項目9社) に並べられたデータを, TR 文で各会社の同じ項目の並びを単位とする4レコードに編集する。

③ OVER TRANS

SECRETARY 終了時にアクティブ・エリアのデータを反転する。

例②を実行後当該コマンドで終わると会社をケースとする横断面データが SECRETARY の "SCWORK1" ファイルに出力される。

〈PUTCROSS 文に関する注意事項と使用例〉

(1) PUTC 文は, アクティブ・エリアの初年度データを1レコードに編集し SECRETARY の "SCWORK1" ファイルに出力する。

SECRETARY セッション中一度以上 PUTC 文を用いると当該文による出力が優先し終了時に "SCWORK1" ファイルに出力される。

(2) PUTC文による出力レコードを一時的に貯えるファイルとして "TMPF3" ファイルを用いる。

(3) 通常は, 本研究所の国際経済経営データベースのデータを SPSS や SAS に引き渡す時でケース数が多い場合に用いる。

## &lt;使用例&gt;

① PERIOD 91 TO 93:

```
GETR #110000,#120000,#210000,#311000,#312000 REPEAT 9501,
9502,9503:
```

```
OPEN TMPF; TMPF3:
```

```
DO WHILE(FLAG(1)=0):
```

```
  RDT REPEAT 5:
```

```
    DO @I=1 TO 5: LET #(@I+5) = #(@I)/10: END:
```

```
  PUTCROSS: END:
```

興銀財務データベースより3社5財務項目を検索し、1991年データに関して各会社5項目の原データと、各項目の1/10の値の5項目を追加した10項目より成るレコード(1ケース)を"TMPF3"ファイルに出力する。

"TMPF3"ファイルに出力されたレコードがあると、SECRETARY終了時に編集して"SCWORK1"ファイル(アクティブエリア)に出力する。

横断面データは、SECRETARYで用いるのみでなく、SPSS等に受渡し分析を行うことができる。その使用例を次に示す。

詳細は、「経済経営研究叢書、経営機械化シリーズ20」、神戸大学経済経営研究所、1989、P.105を参照せよ。

## &lt;例&gt;

PUTCの例題で出力されたデータをSPSSで読む場合。

```
// JOB
```

```
// EXEC SPSS
```

```
//FT08F001 DD DSN=SCWORK1,DISP=SHR
```

```
RUN NAME EXAMPLE-CROSS
```

```
NUMBERD YES
```

```
VARIABLE LIST X1 TO X10
```

INPUT MEDIUM DISK  
 INPUT FORMAT BINARY(56S,10V)  
 N OF CASES 3  
 LIST CASES  
 PEARSON CORR ALL  
 FINISH

《記》

INPUT FORMAT 文の "56S" (スキップ) はそのまま書く。  
 10Vは、読み取る変数(項目)の数に合わせる。

>> TREND (又は TRD)

トレンド曲線

TREND (識別子) [MODEL = 適用モデル [, リードタイム]]  
 [VALUE = ko [, k1]] [LP] [PAGE]

時間を説明変数とする各種のモデルを想定し、最小二乗推定法を用いて時系列データの分析を行う命令文。また推定値は ESTIMATE の名前でベース・エリアにコマンド実行の都度出力される。

• モデル式

- POL1 (多項式一次) .....  $Y_t = a + bt$
- POL2 (多項式二次) .....  $Y_t = a + bt + ct^2$
- POL3 (多項式三次) .....  $Y_t = a + bt + ct^2 + dt^3$
- EXP1 (指数曲線一次) .....  $Y_t = a \cdot b^t$
- EXP2 (指数曲線二次) .....  $Y_t = a \cdot b^t \cdot c^{t^2}$
- EXP3 (指数曲線三次) .....  $Y_t = a \cdot b^t \cdot c^{t^2} \cdot d^{t^3}$
- HYPE (双曲線) .....  $Y_t = \frac{at}{b + t_a}$
- CROW (成長曲線) .....  $Y_t = Ke^{-\frac{t}{\tau}}$

- COMP (ゴンペルツ曲線) .....  $Y_t = Ka^{b^t}$
- LOGI (ロジスティック曲線) .....  $Y_t = \frac{K_0}{1 + me^{-at}}$       $K_0$ : 極限值
- BEST (最適モデル 標準誤差最小)

識別子 (指定しない場合はアクティブ・エリアの1番目):

- ① ベースエリアの変数名。
  - ② # (i); アクティブ・エリアの i 行目のレコードの指定。
  - ③ # Y(j); アクティブ・エリアの j 列目のレコード。
- ・リードタイム: 推定の延長期間を自然数で指定する。
  - ・ $K_0$  (極限值),  $K_1$  (初期値): LOGIモデルに情報を与える。

<使用例>

- ① TRD BASED MODEL = BEST  
データ"BASED"の最適モデルを自動選択して推定。
- ② TRD MODEL = LOGI  
 $k_0, k_1$  を与えていないので三点法により推定。

《記》

プログラムの出所は:

「HSAP統計計算ライブラリー」, HITACマニュアル (800-7-009)。

従って, エラーメッセージ等に関してはマニュアルを参照されたい。

>> UPDATE (又は U)

ベース・レコードの作成, 又は更新

U 名前W (修正開始年) (名前R) (DECIMAL p)

データの作成, または更新を画面で行う文, データはベース・エリアに入れられる。

データの最大サンプル数は, 120迄でありそれを越える場合は, 修正開始年を指定してもう一度 U コマンドを用いる。

- 名前W： 作成，または更新するレコードのコード（第1文字は英字）。  
# (i) でアクティブ・エリアを指定することもできる（i は自然数）。
- 修正開始年： 更新データの画面表示の開始点（西暦年下2桁）を指定。  
位置パラメータであるので，指定する場合は 3語目に書く。
- 名前R： 編集画面に表示するレコード，初期作成時に用いると便利。
- p： 小数点以下の桁数。指定しない場合は画面表示の最初のデータの  
小数点位置が調べられレコードの固有情報として保存される。

《条件・制限》

- 登録済みのデータを更新する場合は，登録期種（年次・四半期・月次）と  
同じ期種モード（PD 文で設定）で行わなければならない。
- 項目数の上限値は次の通り。  
年次（60サンプル），四半期（120サンプル），月次（360サンプル）。

① U CODE1

”CODE1” を作成または更新する。

② U CODE1 CODE2

”CODE2” を基に CODE1 を作成する。

③ IB CDS TO CDT

”CDS” から ”CDT” までのデータ属性を表示。

>> UPDATECAT (又は UC)

カタログの作成，又は更新

UPDATECAT カタログ名 [カタログ名 (新)]

コマンド・プロシージャやコード・リスト等を作成・更新する。

登録済みのカタログの場合は二つのステップを経て更新される。

即ち，① コマンドの送信でその内容が画面に表示さる。

② 表示内容をカーソル・挿入・削除等の機能キーを用いて画面で修正

すればカタログの内容が更新される。

また、カタログ名（新）を書けばその名前で登録される（旧も残る）。

- カタログ名： 第1文字は英字で12文字以内の英数字。
- 関連コマンド： INFO, SAVE, DEL, REN。

#### <使用例>

##### ① UC PROC1

"PROC1" の内容を画面で変更する。

##### ② UC EX1 EX2

"EX1" を画面に表示し修正後の内容を "EX2" で登録する。

#### >> WRITEMT (又はWTM)

アクティブ・エリアのデータの MT 出力

#### WRITEMT

GET系コマンドでアクティブ・エリアに入れられたレコードを SECRETARY  
レコード・フォーマットで磁気テープへ出力する。

出力する期間は PERIOD 文で定義された期間。

### 付 1. 5 行列処理関係

#### 付 1. 5. 1 行列名標の書き方

行列は、次に示す2つの書き方ができる。

##### (a) 行列名標による記述;

行列名のみ、或は添字付き行列名。添字指定は特定の行・列、或は開始点から終了点の連続する範囲として与える。

##### (b) 小行列関数 (SUBMX) の形で書く場合;

この表現の場合は行列の結合機能が有り、且つ行列自身が持つ全ての小行列を表すことができる。

行列演算文で用いる時は上の2つの書き方が可能であるが、分析・出力関連文ではコマンドによって書き方に制限がある。

尚、アクティブ・エリアのデータを行列で表すには行列名の代わりに " # " 記号を用いる。

《行列名標の書き方の例》

例で用いる名標は MX が行列、VEC がベクトルとして登録されているものとする。

転置行列の表示は、名標にアポストロフ( ' )を、逆行列にはダブルアポストロフ( " )を付ける。

〈例〉 VEC',MX (1,3/3,6) ',MX"

(イ) 行列として表されるもの

- MX (行列名のみ)
- MX(1,3) (行列MXの1~3行, 全列の部分行列)
- MX(/2,4) (行列MXの全行, 2~4列の部分行列)
- MX(2,4/3,8) (行列MXの2~4行, 3~8列の部分行列)

(ロ) ベクトルとして表されるもの

- MX(3) (行列MXの3行目の行のベクトル)
- MX(/6) (行列MXの6列目の列のベクトル)
- MX(2/3,8) (MXの2行目, 3~8列のベクトル)
- VEC (ベクトル名のみ)
- VEC(2,7) (ベクトルVECの2~7番目の要素)

(ハ) スカラーとして表されるもの

- @K (スカラー変数(第一文字が@)を書く場合)
- VEC(5) (ベクトルVECの5番目の要素)

- MX(@I/@J) (スカラー変数による添字指定; 行列 MX の @I 行, @J 列の要素)

### 《記》

- アクティブ・エリアを指定する #, 又は #' (転置形) 記号による添字の書き方は, 行列名の書き方に準じる。
- 添字は, 次に示す添字式を書くことができる (BNF 記法);  
 〈添字式〉 ::= 〈定数〉 | 〈変数〉 (+ | - | \*) 〈定数〉  
 カッコ内は演算記号の何れかを選択するものとする。

## 付 1. 5. 2 小行列関数 (SUBMX) について

小行列関数を用いることによって任意の小行列, 任意の行・列ベクトルまたは, 行列の要素をとり出すことができる。

小行列関数は, MATRIX 演算文で用いることの外, 分析或いは出力関連文に書くことができる。

小行列関数を一般形で表わすと:

SUBMX (小行列表記 [ // 小行列表記 ..... // 小行列表記 ])

SUBMX (小行列表記 [ %% 小行列表記 ..... %% 小行列表記 ])

- // (列による結合), %% (行による結合)。
- 小行列表記 (10行列以内); 行列名/行リスト/列リスト/  
 または, 行列名, または, # 記号 (アクティブ・エリア)
- 行列名; 登録済みの行列 (第1文字が英字で7文字以内の文字列)。
- 行リスト・列リスト; 整数値 (数値間はスペース, またはコンマで区切る, 区間で指定する場合は2番目の数値をマイナスにする)。

### 《条件・制限》

行・列リストの制限値は, それぞれ記述項目数で40個, 記述長で80桁。

〈使用例〉

① MAT MX = SUBMX(X/3 -7,9/1,5/)

X行列の3~7, 9行1・5列を MX に入れる。

② MAT @S = SUBMX (X/3/2,-8/)\*V1

X 行列の行ベクトルの要素とベクトル V1 の積。

③ MAT MX = SUBMX(Y//X/2,-4/3,-7/)

行列 Y と X の部分行列を列で結合しその転置行列を MX に入れる。

付 1. 5. 3 固有値及び固有ベクトルを求める関数

(a) EIGNH (x) : 実対称行列。

(b) EIGNQ (x) : 実非対称行列 (一般行列)。

固有値問題  $Xv = \lambda v$

により固有値及び、固有ベクトルを求める組込関数で MATR IX 文で用いられる。

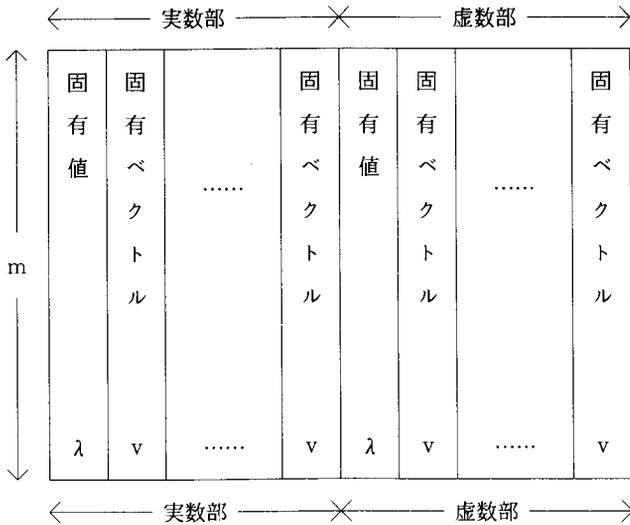
但し、X (被行列), v (固有ベクトル),  $\lambda$  (固有値) とする。

固有値、および固有ベクトルが複素数になる場合は虚数部を含めた行列として出力される。

(a) は、ハウスホルダー法により実対称行列の固有値、および固有ベクトルを求める。

(b) は、QR 法により実非対称行列の固有値、および固有ベクトルを求める。

算出された固有値、および固有ベクトルは下記に示す配置で一つの行列に出力される。



図付 1. 5. 3 固有値・固有ベクトルの配置

付 1. 6 本情報分析システムと統計データベース

データベースは、対象とする分野の情報を網羅的に収集し、関連付け、その生産管理と受益者へのサービスのためのソフトウェアを整備した制御システムとみなすことができる。

データベース自身は 社会環境の変化や利用者のニーズに対しての対応が不可決であり、刻々と変化するデータベースの態様をリアルに利用者に伝達し得るサービス機能をもつ必要がある。

一方利用者側では、データベースの特徴と仕様を見極めると共に、提供している最も新しい情報は何かを把握して使う必要がある。

SECRETARY はこれらの課題にたいして若干の試みを行ったシステムである。

即ち、分析システムとデータベース(\*) との結合を図り、SECRETARY コ

マンドで多分野の収録情報を探索し、所要のデータを抽出し表示するまでの一連の機能を備えている。

使用者は先ずアクセス可能のデータベースの種類（分野）、出所、データの掲載資料等の情報とそのコード体系、利用方法をHELP（又は H）文を用いて確認する。

特定のデータベースの選択は、CFILE 文を用いることでシステムが動的に割り当てる。或は、GET文でデータベース名修飾によるコード記述法を採用することにより任意のデータベース中の特定レコードを検索することもできる。

CFILE文によってシステムに割り当てられたデータベース中の、収録項目・期間等は、DBLIST（又は DBL）文で画面に直ちに表示され同時にアクティブエリア（ACTIVE）に入れられる。又この文ではコード指定を行わなくともブーリアン代数に従った条件による検索を行うこともできる。

次にこれらのコマンドを用いた探索手順例を示す。

HELP DB

登録されている全データベースの一覧表を表示すると共に、所要レコードを探索するための一連の手続き（方法）を使用者に示す。

H DB IFS

IFS データベースに収録されている大分類項目の表示とコード仕様等の説明。

D DB

SECRETARY で割り当てたデータベースの名前、総レコード数等の一覧表。

（\*）昭和49年度文部省試験研究「経営経済情報の自動分析処理システム」

— BEICAシステム— で開発された統計データベース。

その研究プロジェクトを受け継ぎデータの拡充を続け実用に供しているシステムで現在総収録項目数は400万系列に及ぶ。

CFILE KG1A

企業財務データベース（日本興業銀行編集）を割り当てカレントファイルとする（カレントファイルはファイル名による修飾なしで検索できるファイル）。

DBLIST 158

日本（国コード158）に関する収録全項目の検索と表示を行う。

但し、カレントファイルが IFS データベースになっているものとする。

DBL 111 TO 112 AND TEXT=GOLD

米国（111）から英国（112）までの国で、金に関連する項目の表示と検索。

付1. 7 SECRETARY 命令文サマリー

- ・ [ ] で囲まれた単語, 句はオプション指定。
- ・ { } 内は, そのうちの一つを選択する。
- ・ 大文字はコマンド, または, キーワードでありそのまま書く。
- ・ コマンド名のカッコ内は省略形。

機 能	コ マ ン ド	ページ
時系列データの期種変更	AGS [識別子A = ] 識別子B [TO 目的期種] [PREFIX C] コマンド: AGS (累算), AGM (平均), AGK (残高)	
命令文集合の実行	[CALL] プロシージャ名 [C]	
データベースの定義	CFILE データベース名	
表示コードの変更	CHANGECODE(CC) 名前(i) [, 名前(i), ...]	
データの開始時点の変更	CHANGEDATE(CD) {名前 [終了名前]} # 西暦年 [期]}	
表示タイトルの変更	CHANGETITLE (又はCT) i, '文字列'	
ACTIVE のクリアー	CLEARACTIVE(CA)	
列範囲の定義	COL [開始列] [TO 終了列]	
配列演算	COMPUTE(COM) 識別子 = 代数演算式	
データ出力 (その1)	コマンド { (識別子, ..., 識別子) { (TO 識別子) [識別子, ...] } 小行列関数 } [DECIMAL p] [PAGE] [LP] [SKIP n] コマンド: DATA(DT), DATAG(DTG), DATAMPL(DTM), PLOT(PL), SCATTER(SC), SCATTERN(SCN)	
データ出力 (その2)	コマンド [BASIS 基準項目] [CONTROL 欄名] [DECIMAL p] [PAGE] [LP] [SKIP n] コマンド: DATAACROSS(DTC), DTCPLOT	
DBの検索と一覧表	DBLIST(DBL) [コード] [TO コード] [条件式] [LP]	
カタログ等の削除	DELETE(DEL) カタログ名	
BASEデータの削除	DELETE(DEL) BASE.名前B [TO 名前B]	
段落, 行列の削除	DELETE(DEL) USER.段落名	

機 能	コ マ ン ド	ページ
システム状態表示 行列データ出力	DISPLAY(D) 出力指定 DMX { 行列名 [行列名, …… , 行列名] } { 小行列関数 } (DEcimal p) (LP) (PAGE)	
繰り返しの制御	DO @v1=ns TO nt (BY np) DO WHILE (@v2 関係子 c) END	
最小自乗推定による予測	ESTIMATE(EST) { (識別子, …) (TO 識別子) } { 小行列関数 } (WITH ベース名) (PAGE) (LP)	
残差分析	ESTY 小行列関数1, 小行列関数2 (PAGE) (LP)	
表示欄の定義	FIELD(F) 欄名 [, 欄名, …… , 欄名]	
データベース定義	FILE データベース名 [, データベース名, ……]	
レコードの検索	GET(G) コード {コード, ……} (TO コード) (ADD) (索引名 関係子 値, …, 値) (接続子 条件句), ……	
BASE からの読取り	GETBASE(GB) 名前 [, 名前, ……] (TO 名前) (ADD)	
クロスセクション変換	GETCROSS (GETC) GETC コードA {コードA, …… , コードA} REPEAT コードB {コードB, …… , コードB} GETC ¥¥コード {¥¥コード, …… , ¥¥コード} REPEAT ¥¥コード {¥¥コード, …… , ¥¥コード}	
カタログの取込み	GETPUTC (GETP) (ユーザ・カタログ名 = ) システム・カタログ名 (LP)	
繰り返し形検索コマンド (GETREPEAT)	GETR コードA {コードA, …, コードA} REPEAT コードB {コードB, …, コードB} GETR ¥¥コード {¥¥コード, …, ¥¥コード} REPEAT ¥¥コード {¥¥コード, …, ¥¥コード}	
USER からの読取り	GETUSER(GU) 段落名 (段落名, …… , 段落名) (ADD)	
ハードコピー出力	HARDCOPY (HC)	
ヒストグラムの表示	HISTn (又は HISTnC) (a) 系列データを入力する場合	

機 能	コ マ ン ド	ページ
条件文	HIST1~3 [Bコード] [#(n)] (SUBMX (行列名/行リスト/列リスト/)) (LIST (SORT)) (WITHOUT ZERO) (LP) (PAGE) (b) 横断面データを入力する場合 HIST6~8 [コード] (TO コード) (索引名 関係子 値, …… , 値) {接続子 条件句} [(FORM) (n) (V1) (V2)] (LIST (SORT)) (WITHOUT ZERO) (LP) (PAGE)	
ユーザ情報表示	(a) IF v1 関係子 v2 THEN 文: (b) IF v1 関係子 v2 THEN DO: 文1: 文2: …………… END:	
配列演算	INFO(I) 登録名 (TO 登録名) (LP) INFOBASE(IB) 登録名 (TO 登録名) INFOUSER(IU) 登録名 [登録名, …, 登録名] (TO 登録名) (ALL)	
表示コマンド番号の変更	LET 識別子 = 代数演算式	
検索レコードの表示登録	LINE n LIST(L) [条件式] {SAVE {段落名}} [LP] {ACTIVE}	
コマンドの表示 (印刷)	LISTOPTION (LOP) abcd	
最小自乗推定	コマンド { (識別子, ……) (TO 識別子) } LSM等 { (PAGE) (LP) } { 小行列関数 (PAGE) (LP) }	
行列演算文	MATRIX(MAT) 行列 (変数) 名 = 行列演算式	
モードの設定	MODE (MENU メニュー・ファイル名) (ACTIVE)	
新世紀 (2000年対応)	NEWCON (又はNEWCOFF)	
ファイルのオープン	OPEN ファイル名	
画面制御の禁止	NOPAGE	
出力装置の定義	OUTDEV (LP) (LPRINT)	
結果を直ちに高速ブリセッションの終了	OUTHC (OUTH) OVER (又は /*) 改ページ PAGE(PG) (BY 期種) (SPAN i)	
BASEへの書込み	PUTBASE(PB) [名前, …… , 名前]	
DB の指定情報編集出力 (PUTCATALOG)	PC カタログ名 = 欄名 [FORMAT = ' 編集様式'] (WITH 乱数形式, 自然数)	
クロスセクション出力	PUTCROSS(PUTC)	

機 能	コ マ ン ド	ページ
スプレッド表出力	PUTTABLE (PUTT)	
USERへの書込み	PUTUSER(PU) 段落名 (START 収録開始年・期)	
指数乱数	RANDEXP (RNE) 平均 (初期値)	
正規乱数	RANDNOR (RNN) 平均, 標準偏差 (初期値)	
ポアソン乱数	RANDPOIS(RNP) 平均 (初期値)	
一様乱数 (乱数一個生成)	RANDUNIF(RNU) 下限値 TO 上限値 (INIT 初期値) コマンド スカラー変数 コマンド: RNE, RNN, RNP, RNU	
磁気データテープの入力	READMT(RDM)	
逐次読取り (累積)	READSUM(RDS) (REPEAT n)	
逐次読取り (ロード)	READTMPF(RDT) (REPEAT n)	
カタログ名の変更	RANAME(REN) カタログ名 (旧) カタログ名 (新)	
行の範囲の定義	ROW (開始行) (TO 終了行)	
命令文集合の登録	SAVE(S) プロシージャ名	
構造図の作成	SCATTERS(SCS) 小行列関数 SCATTERX(SCX) 小行列関数 (VALUE 最小値, 最大値) (LP) (PAGE)	
行送り	SKIP n	
ACTIVE レコード分類	SORT 欄名 [, 欄名, ……], 欄名 (BY D)	
データ出力 (その1.1)	STATIS(ST) [識別子, ……] (TO 識別子) [識別子, ……] (DECIMAL p)	
ACTIVE データの集計	SUMMATION(SUM) 識別子	
タイトル印刷	TITLE(T) { 'タイトル' メンバー名 (開始行, 終了行) }	
データの転置	TRANSPOSE (REPEAT n) (SAVE 行列名)	
トレンド曲線	TREND [識別子] (MODEL = 適用モデル [, リードタイム]) (TRD) (VALUE = k0 [, k1]) (LP) (PAGE)	
BASE データ作成・更新	U 名前W (修正開始年) (名前R) (DECIMAL p)	
カタログの作成・更新	UPDATECAT(UC) カタログ名 (カタログ名 (新))	
データ磁気テープへ出力	WRITEMT(WTM)	

## 〈付録 2〉 財務分析問題（第5章）実行例

### 付録2目次

	分	析	項	目	頁
【分析例	ANAL01】	財務諸表出力（その1）			
				「データレコードの検索と表出力」	247
【分析例	ANAL02】	財務諸表出力（その2）			248
				「貸借対照表：資産の部、及び負債の部」	
【分析例	ANAL03】	財務諸表出力（その3）		「損益計算書」	249
【分析例	ANAL04】			「収益性分析：資本回転率、売上高利益率」	250
【分析例	ANAL05】			「流動性・安全性分析：流動比率、自己資本比率」	251
【分析例	ANAL06】			「成長性分析：売上高増加率、経常利益増加率（3年平均）」	252
【分析例	ANAL07】	キャッシュフロー分析（その1）			253
				「資本支出最初のみでの時の正味現在価値（NPV）」	
【分析例	ANAL08】	キャッシュフロー分析（その2）			254
				「資本支出最初のみでNPVがプラスになる期を求める」	
【分析例	ANAL09】	キャッシュフロー分析（その3）			255
				「キャッシュフローの正規分布による分析」	
【分析例	ANAL10】	損益分岐点分析（その1）		「データ検索・出力、分析前処理」	256
【分析例	ANAL11】	損益分岐点分析（その2）		「企業群累積値データ出力」	257
【分析例	ANAL12】	損益分岐点分析（その3）			258
				（1）変動費・固定費の算出	
				（2）損益分岐点売上高、安全余裕率（MS比率）	

【主成分分析の例（ANAL13：財務比率の総合1）】	259
「データ検索，各社の各種財務比率行列の作成」	
【主成分分析の例（ANAL14：財務比率の総合2）】	260
「相関行列，固有値，主成分，構造図」	
【主成分分析の例（ANAL15：財務比率の総合3）】	264
「寄与率，分析結果の評価・検証」	
【分析例 ANAL16】 「IFPSによる財務分析」	266
【分析例 ANAL17】 「連結決算子会社の検索とその会社コードのカタログ化」	272
【分析例 ANAL18】 「親会社に対する子会社の構成費の計算出力」	273
【分析例 ANAL19】 「親会社に対する子会社の構成費の計算出力」	274
（ANAL18 続き）	
【分析例 ANAL20】 「単独決算／連結決算対応表出力」	275

【分析例 ANAL01】財務諸表出力（その1）データレコードの検索と表出力  
 《使用カタログ、変数等》NEFA（日経連結決算データベース）

1 >> CFILE NEFA:  
 2 >> PERIOD 90 TO 92:  
 3 >> DBLIST 1301001 TO 1301008:

NO.	CODE	START	TERM.	TYP	AGR	DCP	UPDATE		SERIAL-NAME
1	1301001	84- 1	92- 1	1	0	0		O	KYOKUYO 流動資産合計
2	1301002	84- 1	92- 1	1	0	0		O	KYOKUYO 当座資産合計
3	1301003	84- 1	92- 1	1	0	0		O	KYOKUYO 現金・預金
4	1301004	84- 1	92- 1	1	0	0		O	KYOKUYO 受取手形売掛金
5	1301005	84- 1	92- 1	1	0	0		O	KYOKUYO 非連結子会社関連会社受取手形売掛金
6	1301006	84- 1	92- 1	1	0	0		O	KYOKUYO 有価証券
7	1301007	84- 1	92- 1	1	0	0		O	KYOKUYO 棚卸資産
8	1301008	84- 1	92- 1	1	0	0		O	KYOKUYO 前渡金前払費用

4 >> DATACROSS:

BASIS(1301001 ):KYOKUYO

流動資産合計

NO. コウモク 90.01 円千円 億円

1	KYOKUYO	流動資産合計	59907	1.0000
2	KYOKUYO	当座資産合計	41849	0.6986
3	KYOKUYO	現金・預金	7934	0.1258
4	KYOKUYO	受取手形売掛金	31144	0.5199
5	KYOKUYO	非連結子会社関連会社受取手形	482	0.0080
6	KYOKUYO	有価証券	2689	0.0449
7	KYOKUYO	棚卸資産	13669	0.2282
8	KYOKUYO	前渡金前払費用	2415	0.0403

-- TOTAL --

159689 0.3332

NO. コウモク 91.01 円千円 億円

1	KYOKUYO	流動資産合計	57065	1.0000
2	KYOKUYO	当座資産合計	38159	0.6687
3	KYOKUYO	現金・預金	6977	0.1223
4	KYOKUYO	受取手形売掛金	27964	0.4900
5	KYOKUYO	非連結子会社関連会社受取手形	558	0.0098
6	KYOKUYO	有価証券	2660	0.0466
7	KYOKUYO	棚卸資産	14089	0.2469
8	KYOKUYO	前渡金前払費用	2727	0.0478

-- TOTAL --

150199 0.3290

NO. コウモク 92.01 円千円 億円

1	KYOKUYO	流動資産合計	53589	1.0000
2	KYOKUYO	当座資産合計	30702	0.5729
3	KYOKUYO	現金・預金	5690	0.1062
4	KYOKUYO	受取手形売掛金	20983	0.3916
5	KYOKUYO	非連結子会社関連会社受取手形	2404	0.0449
6	KYOKUYO	有価証券	1625	0.0303
7	KYOKUYO	棚卸資産	17691	0.3301
8	KYOKUYO	前渡金前払費用	2542	0.0474

-- TOTAL --

135226 0.3154

## 【分析例 ANAL02】財務諸表出力(その2)貸借対照表:資産の部、及び負債の部

1 >> CFILE NEFA:			
2 >> PERIOD 90 TO 90:			
3 >> QETR %BSRASSO1 REPEAT 1301:			
4 >> DTC:			
BASIS(1301001 ):KYOKUYO		流動資産合計	
NO.	コウモク	90.01 千円	比率
1	KYOKUYO 流動資産合計	59907	1.0000
2	KYOKUYO 現金・預金	7534	0.1258
3	KYOKUYO 受取手形売掛金	31144	0.5199
4	KYOKUYO 非連結子会社関連会社受取手形	482	0.0080
5	KYOKUYO 有価証券	2689	0.0449
6	KYOKUYO 棚卸資産	13669	0.2282
7	KYOKUYO (▲) 貸倒引当金	-213	-0.0036
8	KYOKUYO 固定資産合計	16879	0.2818
9	KYOKUYO 有形固定資産合計	6487	0.1083
10	KYOKUYO 償却対象有形固定資産	5101	0.0851
11	KYOKUYO 土地・その他	1384	0.0231
12	KYOKUYO 無形固定資産	85	0.0014
13	KYOKUYO 投資その他の資産合計	10305	0.1720
14	KYOKUYO 投資有価証券	6592	0.1100
15	KYOKUYO 非連結子会社関連会社株式・社	1208	0.0202
16	KYOKUYO 長期貸付金	558	0.0093
17	KYOKUYO 非連結子会社関連会社長期貸付	275	0.0046
18	KYOKUYO その他の投資その他の資産	2106	0.0352
19	KYOKUYO (▲) 貸倒引当金	-435	-0.0073
20	KYOKUYO 資産合計	76786	1.2818
	-- TOTAL --	242543	0.2024
5 >> QETR %BSRLIAO1 REPEAT 1301:			
6 >> DTC:			
BASIS(1301027 ):KYOKUYO		流動負債合計	
NO.	コウモク	90.01 千円	比率
1	KYOKUYO 流動負債合計	57055	1.0000
2	KYOKUYO 支払手形買掛金	18024	0.3159
3	KYOKUYO 非連結子会社関連会社支払手形	844	0.0148
4	KYOKUYO 短期借入金・一年内返済の長期	34482	0.6044
5	KYOKUYO 未払金未払費用	1207	0.0212
6	KYOKUYO 未払法人税等	34	0.0006
7	KYOKUYO 短期引当金・未払事業税	422	0.0074
8	KYOKUYO その他流動負債	2038	0.0357
9	KYOKUYO 固定負債合計	10235	0.1794
10	KYOKUYO 長期借入金	8184	0.1434
11	KYOKUYO 退職給与引当金	1929	0.0338
12	KYOKUYO その他長期引当金	121	0.0021
13	KYOKUYO 少数株主特分	66	0.0012
14	KYOKUYO 資本合計	9429	0.1653
15	KYOKUYO 資本金	5664	0.0993
16	KYOKUYO 資本準備金	742	0.0130
17	KYOKUYO 利益準備金	665	0.0117
18	KYOKUYO その他の剰余金	2358	0.0413
19	KYOKUYO 負債・資本合計	76786	1.3458
	-- TOTAL --	230285	0.2124

【分析例 ANAL03】財務諸表出力（その3）損益計算書

```

1 >> CFILE NEFA:
2 >> PERIOD 90 TO 90:
3 >> GETR **PLR01 REPEAT 1301:

4 >> DTC:
BASIS(1301055 ):KYOKUYO          売上高営業収益
NO.      コ  ウ  キ  ク          90.01 千円=      ヒリフ
-----
1 KYOKUYO      売上高営業収益          207862  1.0000
2 KYOKUYO      売上原価営業原価        196281  0.9443
3 KYOKUYO      売上総利益              11581   0.0557
4 KYOKUYO      販売費および一般管理費  10331   0.0497
5 KYOKUYO      営業利益                1249    0.0060
6 KYOKUYO      営業外収益合計          2669    0.0128
7 KYOKUYO      受取利息割引料          412     0.0020
8 KYOKUYO      受取配当金              245     0.0012
9 KYOKUYO      資産処分益              1738    0.0084
10 KYOKUYO     その他営業外収益         273     0.0013
11 KYOKUYO     営業外費用合計          3176    0.0153
12 KYOKUYO     支払利息割引料          2613    0.0126
13 KYOKUYO     その他営業外費用         562     0.0027
14 KYOKUYO     経常利益                 743     0.0036
15 KYOKUYO     特別利益合計            385     0.0019
16 KYOKUYO     特別損失合計            209     0.0010
17 KYOKUYO     税金等調整前当期利益    918     0.0044
18 KYOKUYO     法人税等                 54     0.0003
19 KYOKUYO     持分法による投資損益    354     0.0017
20 KYOKUYO     当期利益                 1212    0.0058

-- TOTAL --          442867  0.1065

5 >> INFO **BSR:
1 **BSRASSO1 #001 #003 #004 #005 #006 #007 #010 #011 #012 #013 #014 #016
  #017 #018 #019 #020 #021 #022 #023 #026
2 **BSREN01  #005 #019 #021 #025 #029 #039 #044 #045 #084 #119 #101 #107
  #108 #109 #110 #116 #117
3 **BSRLIAD1 #027 #028 #029 #030 #032 #033 #034 #035 #036 #038 #040 #041
  #045 #047 #048 #050 #051 #052 #054

6 >> **PLR01:
1 **PLR01    #055 #056 #057 #058 #059 #060 #061 #062 #063 #064 #065 #066
  #069 #070 #071 #074 #080 #081 #085 #086
    
```

【分析例 ANAL04】収益性分析：資本回転率，売上高利益率  
 資本回転率=売上高/使用総資本，売上高利益率=当期利益/売上高(%)  
 《使用カタログ，変数について》

%%RENOSI(項目名リスト)

```

1 >> CFILE NEFA:
2 >> PERIOD 88 TO 92:
3 >> GETR %%RENOSIA REPEAT 6502:
4 >> LET @NOI=NOWD(%%RENOSIA):
5 >>     @SALES=6:
6 >>     @CAPITAL=5:
7 >>     @INCOME=10:
8 >> LET RATIO1=@(SALES) / @(CAPITAL):
9 >> RATIO2=@(INCOME) / @(SALES)*100:

10 >> DTM @(SALES),@(CAPITAL),@(INCOME),RATIO1,RATIO2:
1 6502055 TOSHIBA 売上高営業収益
2 6502054 TOSHIBA 負債・資本合計
3 6502086 TOSHIBA 当期利益
4 RATIO1
5 RATIO2

```

	6502055	6502054	6502086	RATIO1	RATIO2
88.01	3572435	3459133	60711	1.033	1.699
89.01	3800857	4205730	119402	0.904	3.141
90.01	4251953	5181266	131836	0.821	3.101
91.01	4695394	5530370	120852	0.849	2.574
92.01	4722383	5724439	39487	0.825	0.836
TOTAL	21043008	24100928	472288	4.431	11.351
MEAN	4208601	4820185	94458	0.886	2.270
STD.DEV.	463446	858198	37085	0.079	0.886

《演算結果》RATIO1 (資本回転率)、RATIO2 (売上高利益率)

```

11 >> INFO %%RENOSIA:
1 %%RENOSIA #001 #011 #015 #017 #054 #055 #059 #070 #080 #086

```

【分析例 ANAL05】流動性・安全性分析：流動比率，自己資本比率】

流動比率＝流動資産／流動負債，自己資本比率＝資本合計／資産合計（％）  
 《使用カタログ，変数について》

\*\*\*RENOSIB(項目名リスト)

```
1 >> CFILE NEFA:
2 >> PERIOD 88 TO 92:
3 >> GETR ***RENOSIB REPEAT 6502:
```

4 >> DTM:

```
1 6502001 TOSHIBA 流動資産合計
2 6502027 TOSHIBA 流動負債合計
3 6502047 TOSHIBA 資本合計
4 6502026 TOSHIBA 資産合計
```

	6502001	6502027	6502047	6502026	
88.01	2113866	1836119	717938	3459133	
89.01	2723045	2437123	880363	4205730	
90.01	3442153	2756720	1085740	5181266	
91.01	3579999	2924870	1178753	5530370	
92.01	3609070	2936689	1182050	5724439	
TOTAL	15468133	12893521	5044844	24100928	
MEAN	3093626	2578704	1008969	4820185	
STD.DEV.	586680	412936	182162	858198	

```
5 >> LET #NOI=NOWD(***RENOSIB);
6 >> #CASSET=1;
7 >> #CLIABIL=2;
8 >> #CAPITAL=3;
9 >> #ASSETS=4;
10 >> LET RATIO1=#(CASSET) / #(CLIABIL)*100;
11 >> RATIO2=#(CAPITAL) / #(ASSETS)*100;
12 >> DATA RATIO1,RATIO2:
```

1 RATIO1

```
88.01 | 115.13 111.73 124.86 122.40 122.81
```

2 RATIO2

```
88.01 | 20.75 20.93 20.96 21.31 20.65
```

〈財務比率〉 RATIO1 (流動比率)、RATIO2 (自己資本比率)

13 >> INFO \*\*\*RENOSIB:

```
1 ***RENOSIB #001 #027 #047 #026
```

【分析例 ANAL06】成長性分析：売上高増加率、経常利益増加率（3年平均）

\* 売上高増加率 = (当期売上高 / 前期売上高 - 1) × 100

\* 経常利益増加率（3年平均） =  $(\sqrt[3]{\text{当期経常利益} / \text{3年前経常利益}} - 1) \times 100$   
 《使用カタログ、変数》@s（売上高位置変数）、@g（経常利益位置変数）

```

1 >> CFILE NEFA:
2 >> PERIOD 88 TO 92:
3 >> GETR #055,#070 REPEAT 6502:
4 >> LET @NOI=NOWD(##RENO5IC):
5 >>     @S=1:
6 >>     @G=2:
7 >> PD 84:
8 >> LET RATIO1=(#(@S) / #(@S,,-1)-1)*100:
9 >>     RATIO2=((#(@G) / #(@G,,-3))^(1/3) - 1) *100:
<記> "!" はべき乗演算子(16進"5A")
10 >> DTM #(@S),#(@G),RATIO1,RATIO2:
1 6502055 TOSHIBA 売上高営業収益
2 6502070 TOSHIBA 経常利益
3 RATIO1
4 RATIO2
  
```

	6502055	6502070	RATIO1	RATIO2
84.01	2706910	135107	12.740	1.747
85.01	3342824	165738	23.492	16.051
86.01	3372955	130517	0.901	6.948
87.01	3307593	78016	-1.938	-16.727
88.01	3572435	125474	8.007	-12.256
89.01	3800857	234829	6.394	21.626
90.01	4251953	269736	11.868	51.212
91.01	4695394	258853	10.429	27.301
92.01	4722383	114857	0.575	-21.210
TOTAL	33773280	1533127	72.469	74.691
MEAN	3752586	170347	8.052	8.299
STD.DEV.	642668	65449	7.380	22.123

<演算結果> RATIO1（売上高増加率）、RATIO2（3年平均経常利益増加率）

```

11 >> INFO ##RENO5IC:
1 ##RENO5IC #001 #027 #047 #026
  
```

【分析例 ANAL07】キャッシュ・フロー分析 (その1)  
 資本支出最初のみでの時の正味現在価値 (NPV)  
 《使用カタログ, 変数について》

RENO7CFA (キャッシュ・フロー)、RENO7R1 (利子率)、@STY (開始期)、@EDY (終了期)  
 @INVEST (資本支出)、@NPV (正味現在価値)、CUMU (累積値算出関数)  
 RENO7A (PV: 現在価値)、RENO7Q (PVの累積値)

```

1 >> LET @STY=1;
2 >> @EDY=3;
3 >> @INVEST=1000;
4 >> @RR=1;
5 >> PERIOD @STY TO @EDY;

6 >> DO @YY=@STY TO @EDY;
7 >> LET @RR=@RR*RENO7R1(@YY);
8 >> RENO7RR(@YY)=@RR;
9 >> LET RENO7A(@YY)=RENO7CFA(@YY)/@RR;
10 >> END;
11 >> LET RENO7Q=CUMU(RENO7A);
12 >> DTM RENO7CFA,RENO7R1,RENO7RR,RENO7A,RENO7Q:

1 RENO7CFA CASH FLOW DATA-#1.
2 RENO7R1 INTEREST RATIO
3 RENO7RR
4 RENO7A
5 RENO7Q
    
```

	RENO7CFA	RENO7R1	RENO7RR	RENO7A	RENO7Q
1.01	800	1.05	1.050	761.9	761.9
2.01	600	1.04	1.092	549.5	1311.4
3.01	450	1.03	1.125	400.1	1711.4
TOTAL	1850	3.12	3.267	1711.4	3784.7
MEAN	617	1.04	1.089	570.5	1261.6
STD.DEV.	143	0.01	0.031	148.5	389.2

正味現在価値 (@NPV) 算出と結果

```

13 >> LET @NPV=RENO7Q(@EDY)-@INVEST;
14 >> D SCAL @NPV:

@NPV 711.442 7.114419E+02
    
```

【分析例 ANAL08】キャッシュ・フロー分析(その2)  
 資本支出最初のみでNPVがプラスになる期を求める  
 《使用カタログ、変数》

RENO7CFA (キャッシュ・フロー)、RENO7R1 (利率)  
 SINVEST (資本支出)、BNPV (正味現在価値)  
 RENO7A (PV: 現在価値)、RENO7P (NPVの動き)

```

1 >> LET @STY= 1;
2 >> @EDY= 3;
3 >> @INVEST=1000;
4 >> @NPV=-@INVEST;
5 >> @PV=0;
6 >> @RR=1;
7 >> RENO7P=0;
8 >> PERIOD @STY TO @EDY;

9 >> DO @YY=@STY TO @EDY WHILE(@NPV<0);
10 >> LET @RR=@RR*RENO7R1(@YY);
11 >> RENO7RR(@YY)=@RR;
12 >> LET @PV=RENO7CFA(@YY)/@RR;
13 >> RENO7A(@YY)=@PV;
14 >> LET @NPV=@NPV+@PV;
15 >> RENO7P(@YY)=@NPV;
16 >> END;
17 >> DTM RENO7CFA,RENO7R1,RENO7RR,RENO7A,RENO7P;
1 RENO7CFA CASH FLOW DATA-#1.
2 RENO7R1 INTEREST RATIO
3 RENO7RR
4 RENO7A
5 RENO7P
  
```

	RENO7CFA	RENO7R1	RENO7RR	RENO7A	RENO7P
1.01	800	1.05	1.050	761.9	-238.09
2.01	600	1.04	1.092	549.5	311.36
3.01	450	1.03	1.125	400.1	0.00
TOTAL	1850	3.12	3.267	1711.4	73.26
MEAN	617	1.04	1.089	570.5	24.42
STD.DEV.	143	0.01	0.031	148.5	224.98

\* NPVがプラスになる期(@YY)、及びNPVの値

```

18 >> LET @YY=@YY-1;
19 >> D SCAL @YY, @NPV;
      @YY          2.000    2.000000E+00
      @NPV        311.356    3.113562E+02
  
```

【分析例 ANAL09】キャッシュ・フロー分析 (その3)  
 キャッシュ・フローの正規分布による分析

RENO7F (CF), @R (利率), @INVEST (資本支出)

```

1 >> LET @STY=1;
2 >> @EDY=5;
3 >> @INVEST= 600;
4 >> @NPVE=-@INVEST;
5 >> @R=1.03;
6 >> PERIOD @STY TO @EDY;
7 >> LET @NPVS=0;
8 >> W2=0;
9 >> @B RENO7FA TO RENO7FC;
10 >> DO @K=1 TO 3;
11 >> LET @AV=MEAN(@(@K));
12 >> @(@COUNT+1)=(@(@K)-@AV)*(@(@K)-@AV);
13 >> END;
14 >> LET W1=POW(@R);
15 >> DO @K=1 TO 3;
16 >> LET W2(@K)=EXPCT(@(@K),RENO7PW);
17 >> @NPVE=@NPVE+W2(@K)/W1(@K);
18 >> LET @NPVS=@NPVS+EXPCT(@(@K+3),RENO7PW) / @R!(@K*2);
19 >> END;
20 >> LET @NPVS=SQRT(@NPVS);
21 >> DTM @(1) TO @(4),RENO7PW;
1 RENO7FA   キャッシュ・フロー (第1期ファンド) .
2 RENO7FB   キャッシュ・フロー (第2期ファンド) .
3 RENO7FC   キャッシュ・フロー (第3期ファンド) .
4 @(@COUNT+1)
5 RENO7PW   確率.
    
```

	RENO7FA	RENO7FB	RENO7FC	@(@COUNT+1)	RENO7PW
1.01	600	550	500	40000	0.10
2.01	500	450	400	10000	0.25
3.01	400	350	300	0	0.30
4.01	300	250	200	10000	0.25
5.01	200	150	100	40000	0.10
TOTAL	2000	1750	1500	100000	1.00
MEAN	400	350	300	20000	0.20
STD.DEV.	141	141	141	16733	0.08

22 >> DATA W1,W2:

```

1 W1
1.01 | 1.030 | 1.061 | 1.093 | 1.126 | 1.159

2 W2
1.01 | 400.00 | 350.00 | 300.00 | 0.00 | 0.00

23 >> LET @Z=(0-@NPVE)/@NPVS;
24 >> @P=0.5+TBLN(@Z);
25 >> D SCAL,@NPVE,@NPVS,@Z,@P;
      @NPVE      392.800      3.928000E+02
      @NPVS      186.256      1.862562E+02
      @Z         -2.109      -2.108923E+00
      @P         0.017      1.747537E-02
    
```

【分析例 ANAL10】損益分岐分析(その1): データ検索・出力, 分析前処理

\* 損益分岐点 = 固定費 ÷ (1 - 変動費 / 売上高)

\* 安全余裕率 = (当期売上高 - 損益分岐点売上高) / 当期売上高  
 《使用カタログ, 変数》

KN1A (興銀財務データベース名, 上場1部)

##RENO6IA (検索項目コード・リスト), ##RENO6KA (検索会社リスト)

```

1 >> CFILE KN1A:
2 >> PERIOD 88 TO 92:
3 >> GETR ##RENO6IA REPEAT ##RENO6KA:
50 9232317185 (株)パスコ: 販売費および一般管理
  変動費・固定費算出のための関連データの検索
4 >> OPEN THPF:
5 >> READTHPF:
  
```

各々の項目について検索した全企業の累積をとる

```

6 >> DO WHILE(FLAG(1)=0):
7 >> READSUM:
8 >> END:
9 >> LIST:

1  9201312000  日本航空: 売上原価
2  9201413010  日本航空: 経費 (うち外注加工費)
3  9201413020  日本航空: 経費 (うち動力燃料費)
4  9201317010  日本航空: 販売手数料
5  9201317020  日本航空: 荷造運搬費
6  9201317040  日本航空: その他販売費
7  9201317050  日本航空: 特許使用料
8  9201317060  日本航空: 貸倒引当金繰入額および
9  9201317155  日本航空: 租税公課 (うち事業税
10 9201317160  日本航空: 物品税
11 9201317183  日本航空: 販売費および一般管理
12 9201317185  日本航空: 販売費および一般管理
13 9201412000  日本航空: 労務費
14 9201413000  日本航空: 経費
15 9201311000  日本航空: 売上高
16 9201319000  日本航空: 営業損益
17 9201321010  日本航空: 受取利息・配当金
18 9201322010  日本航空: 支払利息・割引料
19 9201322020  日本航空: 社債発行差金償却及び

## OUTPUT RECORDS= 19
10 >> INFO ##RENO6:
1  ##RENO6IA #312000 #413010 #413020 #317010 #317020 #317040 #317050 #317
  060 #317155 #317160 #317183 #317185 #412000 #413000 #311000
  #319000 #321010 #322010 #322020
2  ##RENO6KA 9201.9202.9232:
11 >> LET @ITEMS=@COUNT:
12 >> D SCAL @ITEMS:
      @ITEMS          19.000    1.900000E+01
  
```

損益分岐点分析 (その2 ANAL11) 企業群累積値データ出力

1 >> PERIOD 90 TO 92:  
 2 >> DTM # (1), # (2), # (4), # (8), # (9), # (11) TO # (19):

	9201312000	9201413010	9201317010	9201317060	9201317155
1 9201312000	日本航空:				
2 9201413010	日本航空:				
3 9201317010	日本航空:				
4 9201317060	日本航空:				
5 9201317155	日本航空:				
90.01	1310904	135992	138578	443	7079
91.01	1467273	156893	146807	241	3335
92.01	1530420	176444	159250	117	2204
TOTAL	4308597	469329	444635	801	12618
MEAN	1436199	156443	148212	267	4206
STD.DEV.	92268	16517	8498	134	2083

	9201317183	9201317185	9201412000	9201413000	9201311000
6 9201317183	日本航空:				
7 9201317185	日本航空:				
8 9201412000	日本航空:				
9 9201413000	日本航空:				
10 9201311000	日本航空:				
90.01	4103	2992	307387	680493	1786653
91.01	3996	3457	328887	743593	1914896
92.01	2756	3892	351392	812428	1980981
TOTAL	10855	10341	987666	2236514	5682532
MEAN	3618	3447	329222	745505	1894177
STD.DEV.	611	367	17966	53877	80682

	9201319000	9201321010	9201322010	9201322020
11 9201319000	日本航空:			
12 9201321010	日本航空:			
13 9201322010	日本航空:			
14 9201322020	日本航空:			
90.01	115956	33669	56299	7007
91.01	58041	48711	65512	1113
92.01	33208	43236	80752	2474
TOTAL	207205	125616	202563	10594
MEAN	69068	41872	67521	3531
STD.DEV.	34670	6216	10084	2520

## 【分析例 ANAL12】損益分岐点分析(その3)

(1) 変動費, 固定費の算出 (2) 損益分岐点売上高, 安全余裕率 (MS比率)

\* 損益分岐点 = 固定費 ÷ (1 - 変動費 / 売上高)

\* 安全余裕率 = (当期売上高 - 損益分岐点売上高) / 当期売上高

```

1 >> PD 88 TO 92:
2 >> LET @ANSV=@ITEMS+1:
3 >>      #(@ANSV)=0:

4 >> DO @I=1 TO 12:
5 >> LET #(@ANSV)=#(@ANSV)+#(@I):
6 >> END:
7 >> LET #(@ANSV)=#(@ANSV)-#(13)-#(14):
8 >>      COSTV=#(@ANSV):
9 >>      COSTF=#(15)-( #(16)+#(17)-#(18)-#(19))-#(@ANSV):
10 >>      BEP=COSTF / (1 - COSTV / #(15)):
11 >>      MSR=(#(15)-BEP) / #(15):
12 >> DTM COSTV,COSTF,BEP,#(15),MSR:

1 COSTV
2 COSTF
3 BEP
4 SALES      売上高合計
5 MSR

```

	COSTV	COSTF	BEP	SALES	MSR
88.01	495930.00	876922.00	1349489.00	1416205	0.04711
89.01	523575.00	973972.00	1466412.00	1559126	0.05947
90.01	612211.00	1088125.00	1655339.00	1786655	0.07350
91.01	709522.00	1165247.00	1851148.00	1914896	0.03329
92.01	711263.00	1276500.00	1991561.00	1980981	-0.00534
TOTAL	3052501.00	5380766.00	8313949.00	8657863	0.20802
MEAN	610500.19	1076153.00	1662789.00	1731572	0.04160
STD.DEV.	90158.56	140250.63	236796.31	213565	0.02698

## 〈演算結果〉

COSTV: 変動費, COSTF: 固定費, BEP: 損益分岐点, MSR: 安全余裕率

```

13 >> D SCAL @COUNT:
          @COUNT          20.000    2.000000E+01

```

【成分分析の例 ANAL13：財務比率の総合1】  
データ検索，各社の各種財務比率行列の作成

```

1 >> CFILE KN1A:
2 >> GETR **KGIT01 REPEAT **KGC001:
50 6502317130 (株)東芝: 減価償却費
100 6504221000 富士電機: 資本金
150 6506130040 (株)安川電機: 開発費試験研究費
200 6508120000 (株)明電舎: 固定資産
250 6510395000 日本電気精器: 当期末処分利益金

3 >> PERIOD 93 TO 94:
4 >> LET BK=1:
5 >> OPEN TMPF:
6 >> DO WHILE(FLAG(1)=0):
7 >> READTMPF:
8 >> LET #(31)=#(17)/#(11):
9 >> #(32)=#(23)/#(17):
10 >> #(41)=#(4)/(#(11)-#(12)):
11 >> #(42)=#(17)/#(4):
12 >> #(36)=#(17)/#(28):
13 >> #(33)=#(1)/#(13):
14 >> #(35)=#(24)/#(11):
15 >> #(34)=#(2)/#(13):
16 >> #(38)=#(17)/#(3):
17 >> IF BK=1 THEN MAT MMX=#(31,35/1):
18 >> MAT MMX=#(31,35/1):
19 >> IF BK>1 THEN MAT MM2=#(31,35/1):
20 >> IF BK>1 THEN MAT MMX=SUBMX(MMX//MM2):
21 >> LET BK=BK+1:
22 >> END:
23 >> MAT MM2=#(31,35/1):
24 >> MAT MMX=SUBMX(MMX//MM2):
25 >> DMX SUBMX(MMX/1,-5/1,-5/1):
    
```

	COL# 1	COL# 2	COL# 3	COL# 4	COL# 5
ROW# 1	0.9769	0.8702	1.013	0.9810	1.003
ROW# 2	0.0152	0.0123	0.008	0.0206	-0.022
ROW# 3	1.6148	1.1575	1.441	1.3137	1.215
ROW# 4	1.0467	0.6409	0.872	0.8371	0.873
ROW# 5	0.0200	0.0151	0.013	0.0146	-0.028
TOTAL	3.6736	2.6960	3.347	3.1671	3.042
MEAN	0.7347	0.5392	0.669	0.6334	0.608
STD.DEV.	0.6259	0.4592	0.569	0.5260	0.529

```

26 >> IU MMX MM2:
NO NAME UPDATE ROW COL.
1 MMX@ 951004-1314 5 10 PROD.-DATE:950403, TIME:163846
2 MM2@ 951004-1314 5 1 PROD.-DATE:950405, TIME:202130
27 >> I **KGIT:
1 **KGIT01 #110000 #111000 #112000 #120000 #121000 #122000 #123000
#123070 #130000 #130040 #190000 #210000 #211000 #212000
#220000 #221000 #311000 #312000 #312050 #316000 #317000
#317130 #319000 #329000 #390000 #395000 #502040 #502010
28 >> **KGC001:
1 **KGC001 6501 6502 6503 6504 6505 6506 6507 6508 6510 6513
    
```

【成分分析の例 ANAL 14 : 財務比率の総合2】  
 相関行列, 固有値, 主成分, 構造図

```

1 >> LET @NPC = 5;
2 >> @NPC1=@NPC+1;

3 >> D SCAL @NPC,@NPC1:
          @NPC          5.000    5.000000E+00
          @NPC1         6.000    6.000000E+00

4 >> MAT MD1 = NORD(MMX)';
5 >> MXR = CORX(MD1);
6 >> MXX = EIGNH(MXR);
7 >> MXS = SDEV(MD1' * MXX(/2,@NPC1));
8 >> MW2 = MXX(/2,@NPC1) * DIAG(1/MXS);
9 >> MD2 = MD1' * MW2;
10 >> MST = MW2' * MXR;
11 >> DMX MXR:

```

PROD.-DATE:851226, TIME:111120

	MXR:C001	MXR:C002	MXR:C003	MXR:C004	MXR:C005
ROW#001	1.000	0.277	-0.504	-0.557	0.281
ROW#002	0.277	1.000	-0.105	-0.226	0.898
ROW#003	-0.504	-0.105	1.000	0.966	0.157
ROW#004	-0.557	-0.226	0.966	1.000	-0.010
ROW#005	0.281	0.898	0.157	-0.010	1.000
TOTAL	0.498	1.844	1.513	1.174	2.326
MEAN	0.100	0.369	0.303	0.235	0.465
STD.DEV.	0.578	0.503	0.594	0.635	0.407

12 >> DMX SUBMX(MXX/1,-5/1,-5/):

	COL# 1	COL# 2	COL# 3	COL# 4	COL# 5
ROW# 1	2.521	-0.4854	-0.01411	0.8594	0.1533
ROW# 2	1.853	-0.3545	0.57010	-0.3156	0.6261
ROW# 3	0.547	0.5143	0.38764	0.2914	-0.1775
ROW# 4	0.065	0.5585	0.28959	0.2773	0.4197
ROW# 5	0.014	-0.2498	0.66382	-0.0019	-0.6139
TOTAL	5.000	-0.0169	1.89704	1.1107	0.4077
MEAN	1.000	-0.0034	0.37941	0.2221	0.0815
STD.DEV.	1.009	0.4472	0.23675	0.3881	0.4397

13 >> DMX MW2;  
 PROD.-DATE:851224, TIME:194843

	MW2:C001	MW2:C002	MW2:C003	MW2:C004	MW2:C005
ROW:001	-0.306	-0.010	1.162	0.399	0.40
ROW:002	-0.223	0.419	-0.427	2.446	2.05
ROW:003	0.324	0.285	0.394	-0.694	3.85
ROW:004	0.332	0.213	0.375	1.640	-5.06
ROW:005	-0.157	0.488	-0.003	-2.399	-2.96
TOTAL	-0.011	1.394	1.501	1.593	0.27
MEAN	-0.002	0.279	0.300	0.319	0.05
STD.DEV.	0.282	0.174	0.525	1.718	3.82

14 >> MD2;  
 PROD.-DATE:851226, TIME:111335

	MD2:C001	MD2:C002	MD2:C003	MD2:C004	MD2:C005
ROW:001	-0.247	0.862	0.938	-0.131	0.333
ROW:002	-0.529	0.174	-1.693	-2.309	0.154
ROW:003	-0.637	0.403	1.281	-0.564	1.055
ROW:004	-0.747	0.523	0.389	0.245	0.517
ROW:005	-0.291	-0.893	1.428	0.013	-0.619
ROW:006	1.467	-0.098	-0.008	0.036	-1.099
ROW:007	-0.725	0.659	-1.456	2.018	0.669
ROW:008	-0.595	0.421	-0.268	0.291	-2.352
ROW:009	-0.058	-2.649	-0.455	0.328	0.592
ROW:010	2.362	0.598	-0.157	0.075	0.750
TOTAL	-0.000	0.000	0.000	0.000	-0.000
MEAN	-0.000	0.000	0.000	0.000	-0.000
STD.DEV.	1.000	1.000	1.000	1.000	1.000

15 >> DMX MST';  
 PROD.-DATE:851226, TIME:111502

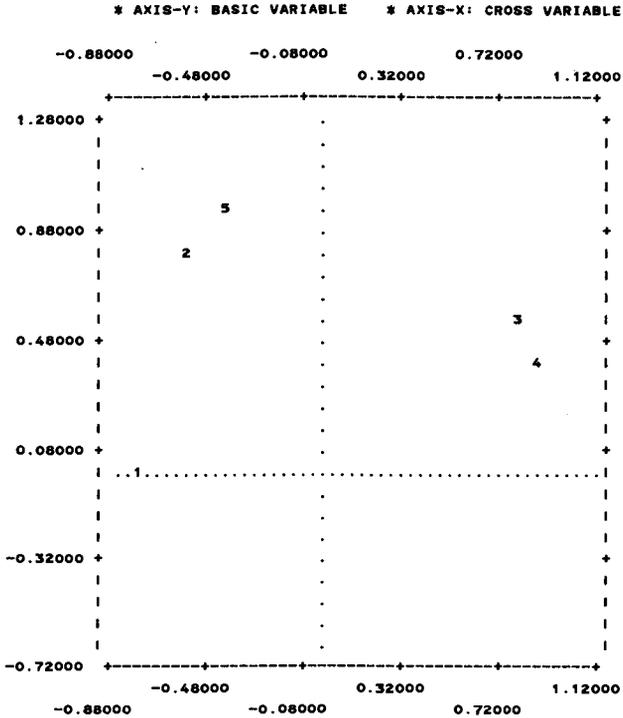
	MST:C001	MST:C002	MST:C003	MST:C004	MST:C005
ROW:001	-0.771	-0.019	0.636	0.0392	0.0054
ROW:002	-0.563	0.776	-0.233	0.1602	0.0281
ROW:003	0.816	0.528	0.216	-0.0454	0.0802
ROW:004	0.887	0.394	0.205	0.1074	-0.0694
ROW:005	-0.397	0.904	-0.001	-0.1371	-0.0406
TOTAL	-0.027	2.582	0.822	0.1044	0.0038
MEAN	-0.005	0.516	0.164	0.0209	0.0008
STD.DEV.	0.710	0.322	0.287	0.1125	0.0524

16 >> DMX HXS:  
 PROD.-DATE:851224, TIME:114752

	MXS:C001	MXS:C002	MXS:C003	MXS:C004	MXS:C005
ROW:001	1.588	1.361	0.7398	0.2559	0.1171
TOTAL	1.588	1.361	0.7398	0.2559	0.1171
MEAN	1.588	1.361	0.7398	0.2559	0.1171
STD.DEV.	0.000	0.000	0.0000	0.0000	0.0000

17 >> SCX SUBHX(MST/2,1/1,-5/) VALUE -0.5,1:

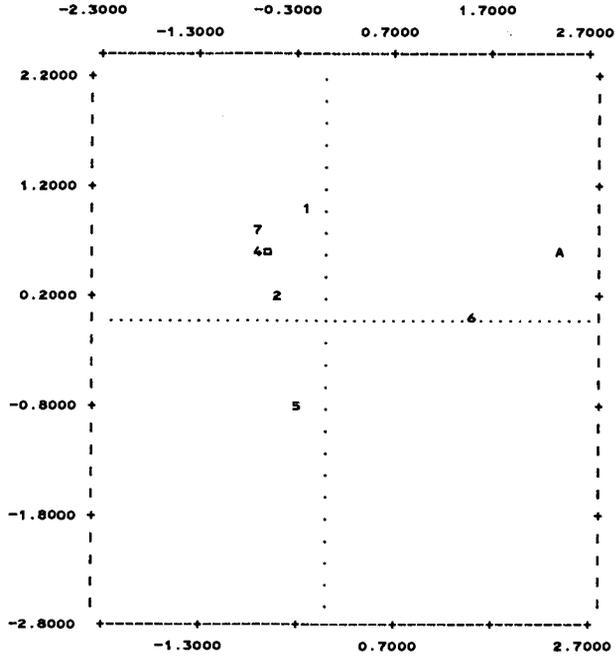
BASIS(ROW# 1 ):  
 # 1(ROW# 2 ):



18 >> SCX SUBMX(MD2/1,-10/2.1)':

BASIS(COL# 1 ):  
# 1(COL# 2 ):

\* AXIS-Y: BASIC VARIABLE \* AXIS-X: CROSS VARIABLE



19 >> IU MD1,MD2,MST,MXR,MXS,MXH,MW2:

NO	NAME	UPDATE	ROW	COL	
1	MD1#	951004-1315	5	10	PROD.-DATE:851226, TIME:111119
2	MD2#	951004-1315	10	5	PROD.-DATE:851226, TIME:111335
3	MST#	951004-1315	5	5	PROD.-DATE:851226, TIME:111502
4	MXR#	951004-1315	5	5	PROD.-DATE:851226, TIME:111120
5	MXS#	951004-1315	1	5	PROD.-DATE:851224, TIME:114752
6	MXH#	951004-1315	5	6	TEST OF FUNCTION. 1994.6.6
7	MW2#	951004-1315	5	5	PROD.-DATE:851224, TIME:194843

## 【成分分析の例 ANAL 15 : 財務比率の総合3】

寄与率, 分析結果の評価・検証

```

1 >>  MAT MW1 = CUMU(MXX(/1));
2 >>  MW2 = MW1 / @NPC * 100;
3 >>  MX = MXX(/2,@NPC1)' * MXX(/2,@NPC1);
4 >>  MW3 = MXX(/2,@NPC1)' * MXR * MXX(/2,@NPC1);
5 >>  MY1 = MXR * MXX(/2,@NPC1);
6 >>  MY2 = MXX(/2,@NPC1) * DIAG(MXX(/1)');
7 >>  DMX MW1,MW2;

```

PROD.-DATE:851224, TIME:194842

PROD.-DATE:851224, TIME:194843

	MW1:COO1	MW2:COO1
ROW:001	2.52	50.4
ROW:002	4.37	87.5
ROW:003	4.92	98.4
ROW:004	4.99	99.7
ROW:005	5.00	100.0
TOTAL	21.80	436.0
MEAN	4.36	87.2
STD.DEV.	0.95	19.0

8 &gt;&gt; MX DECIMAL 4:

KMAST

	MX:COO1	MX:COO2	MX:COO3	MX:COO4	MX:COO5
ROW:001	1.0000	0.0000	-0.0000	0.0000	0.0000
ROW:002	0.0000	1.0000	0.0000	0.0000	0.0000
ROW:003	-0.0000	0.0000	1.0000	0.0000	-0.0000
ROW:004	0.0000	0.0000	0.0000	1.0000	-0.0000
ROW:005	0.0000	0.0000	-0.0000	-0.0000	1.0000
TOTAL	1.0000	1.0000	1.0000	1.0000	1.0000
MEAN	0.2000	0.2000	0.2000	0.2000	0.2000
STD.DEV.	0.4000	0.4000	0.4000	0.4000	0.4000

9 >> DMX MW3 DECIMAL 3:  
 PROD.-DATE:851224, TIME:200959

	MW3:C001	MW3:C002	MW3:C003	MW3:C004	MW3:C005
ROW:001	2.521	-0.000	0.000	-0.000	0.000
ROW:002	-0.000	1.853	-0.000	0.000	0.000
ROW:003	0.000	-0.000	0.547	-0.000	0.000
ROW:004	-0.000	0.000	-0.000	0.066	-0.000
ROW:005	-0.000	0.000	0.000	-0.000	0.014
TOTAL	2.521	1.853	0.547	0.066	0.014
MEAN	0.504	0.371	0.109	0.013	0.003
STD.DEV.	1.008	0.741	0.219	0.026	0.005

10 >> DMX MY1:  
 PROD.-DATE:851221, TIME:183452

	MY1:C001	MY1:C002	MY1:C003	MY1:C004	MY1:C005
ROW:001	-1.223	-0.026	0.4704	0.0100	0.00064
ROW:002	-0.894	1.056	-0.1727	0.0410	0.00329
ROW:003	1.296	0.718	0.1595	-0.0116	0.00939
ROW:004	1.408	0.537	0.1518	0.0275	-0.00812
ROW:005	-0.630	1.230	-0.0010	-0.0402	-0.00475
TOTAL	-0.043	3.515	0.6080	0.0267	0.00044
MEAN	-0.009	0.703	0.1216	0.0053	0.00009
STD.DEV.	1.127	0.439	0.2125	0.0288	0.00613

11 >> MY2:  
 PROD.-DATE:851221, TIME:183453

	MY2:C001	MY2:C002	MY2:C003	MY2:C004	MY2:C005
ROW:001	-1.223	-0.026	0.4704	0.0100	0.00064
ROW:002	-0.894	1.056	-0.1727	0.0410	0.00329
ROW:003	1.296	0.718	0.1595	-0.0116	0.00939
ROW:004	1.408	0.537	0.1518	0.0275	-0.00813
ROW:005	-0.630	1.230	-0.0010	-0.0402	-0.00475
TOTAL	-0.043	3.515	0.6080	0.0267	0.00044
MEAN	-0.009	0.703	0.1216	0.0053	0.00009
STD.DEV.	1.127	0.439	0.2125	0.0288	0.00613

12 >> IU MW1 TO MW3, HX, HXR, MY1, MY2:

NO	NAME	UPDATE	ROW	COL.
1	MW1#	951004-1315	5	1 PROD.-DATE:851224, TIME:194842
2	MW2#	951004-1315	5	1 PROD.-DATE:851224, TIME:194843
3	MW3#	951004-1315	5	5 PROD.-DATE:851224, TIME:200959
4	HX#	951004-1315	5	5
5	HXX#	951004-1315	5	6 TEST OF FUNCTION. 1996.6.6
6	HXR#	951004-1315	5	5 PROD.-DATE:851226, TIME:111120
7	MY1#	951004-1315	5	5 PROD.-DATE:851221, TIME:183452
8	MY2#	951004-1315	5	5 PROD.-DATE:851221, TIME:183453

【分析例 ANAL16】 IFPSによる財務分析

READY

IFPS..... I F P S呼出コマンド

IFPS/PLUS 4.0.2

INTERACTIVE FINANCIAL PLANNING SYSTEM

OPTIMUM 4.2

ENTER NAME OF FILE CONTAINING MODELS AND REPORTS

? IFPSCMD .....カタログファイル指定

FILE IFPSCMD PROCESSED

READY FOR EXECUTIVE COMMAND

? CATALOG .....カタログのリスト

NAME	TYPE	DATE	TIME	LINES	VERSION
ANALBP1	MODEL	05/08/95	19:20:24	35	SAVED
POLYFIT	MODEL	08/12/92	14:50:50	4	SAVED
REPORT1	REPORT	08/25/86	16:45:34	35	SAVED
INCOME	MODEL	08/25/86	15:27:58	28	SAVED

READY FOR EXECUTIVE COMMAND

? MODEL ANALBP1 .....モデルの選択

MODEL ANALBP1 READY FOR EDIT, LAST LINE IS 440

? LIST.....プログラムのリスト

MODEL ANALBP1 VERSION OF 05/08/95 19:20

100 \* 財務分析の例 (EXAMPLE OF FINANCIAL ANALYSIS)

110 \* 新製品製品化のための設備投資によって貸借対照表・損益計算書の

120 \* 主要項目の動きを5期にわたって予測する。続いて、投資計画の入力

130 \* 計数値を変えた場合にどのような影響がでるか分析 (WHAT IF), 収

140 \* 益の目標値を達成するための売上高、即ち確保しなければならない生

150 \* 産性の分析 (GOAL SEEKING)KING) を行う。

160 COLUMNS 0-5

170 \* 売上高

180 SALES = 2921473, PREVIOUS SALES \* 1.03

190 \* 売上原価 = 期首商品棚卸高 - 期末商品棚卸高 + 当期商品仕入高 + 物品税

200 \* COST OF GOODS = INVENTS - INVENTE + GOODS + TAXM

210 COST OF GOODS = 2127607, PREVIOUS COST OF GOODS \* 1.05

220 \* 変動費 = 原材料費 + 外注加工費 + 電力・ガス・水道 + 運賃 + 特許料

230 \* VARCOST = MATER + OUTORDER + EGW + FARE + PATENT

240 VARCOST = 1151743, PREVIOUS VARCOST \* 1.03

250 \* 固定費 = 売上高 - 変動費 - 利払後事業損益 (営業損益 + 受取利息・配当金

260 \* FIXCOST = SALES - VARCOST - AFTGAIN (OPEGAIN + INTERESTR)

270 FIXCOST = SALES - VARCOST

280 \* 損益分岐点 = 固定費 ÷ (1 - 変動費 / 売上高)

290 BEP = FIXCOST / (1 - VARCOST / SALES)

300 \* 売上純利益 = 売上高 - 売上原価

310 GROSSMGN = SALES - COST OF GOODS

320 \* 税引前利益 = 売上純利益 - 一般管理費 - 支払利息

330 \* BEFGAIN = GROSSMGN - OPER EXPENSE - INTERESTE

340 BEFGAIN = GROSSMGN - 617408

350 \* 定額法により減価償却額を求める

360 \* 減価償却関数 : STLINE DEPR

370 \* (資産原価, 残存簿価率, 耐用年数, 期間償却額, 簿価, 累積償却額)

380 STLINE DEPR (INVESTMENT, 0.1, 5, DEPRECIATION, BOOK VALUE, CUM DEPRE)

390 INVESTMENT = 300000, 0

400 \* 内部利益率関数 : IRR (期間収入, 期間投資額)

410 DEF = IRR(NET CASH FLOW,INVESTMENT)

420 \* ネットキャッシュフロー=税引前利益-法人税+減価償却額

430 NET CASH FLOW=BEFGAIN-TAXC+DEPRECIATION

440 TAXC=12345 , PREVIOUS TAXC \* 1.01

END OF MODEL

?SOLVE.....モデルの実行

MODEL ANALBP1 VERSION OF 05/08/95 19:20-6 COLUMNS 14 VARIABLES

ENTER SOLVE OPTIONS

? ALL

0 1 2 3 4 5

財務分析の例 (EXAMPLE OF FINANCIAL ANALYSIS)

新製品製品化のための設備投資によって貸借対照表・損益計算書の主要項目の動きを5期にわたって予測する。続いて、投資計画の入力計数値を変えた場合にどのような影響がでるかの分析 (WHAT IF)、収益の目標値を達成するための売上高、即ち確保しなければならない生産性の分析 (GOAL SEEKING) を行う。

売上高

SALES 2921473 3009117 3099391 3192372 3288144 3386788

売上原価=期首商品棚卸高-期末商品棚卸高+当期商品仕入高+物品税

COST OF GOODS=INVENTS-INVENTE +GOODS +TAXM

COST OF GOODS 2127607 2233987 2345687 2462971 2586120 2715426

変動費=原材料費+外注加工費+電力・ガス・水道+運賃+特許料

VARCOST=MATER+OUTORDER+EGW +FARE +PATENT

VARCOS 1151743 1186295 1221884 1258541 1296297 1335186

固定費=売上高-変動費-利払後事業損益 (営業損益+受取利息・配当金

FIXCOST=SALES-VARCOST-AFTGAIN (OPEGAIN +INTERESTR)

FIXCOST            1769730 1822822 1877507 1933832 1991847 2051602

損益分岐点=固定費÷(1-変動費/売上高)

BEP                    2921473 3009117 3099391 3192372 3288144 3386788

売上純利益=売上高-売上原価

GROSSMGN        793866 775130 753704 729401 702024 671362

税引前利益=売上純利益 - 一般管理費-支払利息

BEFGAIN        = GROSSMGN - OPER EXPENSE - INTERESTE

BEFGAIN            176458 157722 136296 111993 84616 53954

定額法により減価償却額を求める

減価償却関数:    S T L I N E   D E P R

(資産原価, 残存簿価率, 耐用年数, 期間償却額, 簿価, 累積償却額)

DEPRECIATION        54000 54000 54000 54000 54000 0

BOOK VALUE            246000 192000 138000 84000 30000 30000

CUM DEPRE            54000 108000 162000 216000 270000 270000

INVESTMENT            300000 0 0 0 0 0

内部利益率関数:    I R R (期間収入, 期間投資額)

DEF                    .2559 .4599 .5443 .5808 .5876

ネットキャッシュフロー = 税引前利益-法人税+減価償却額

NET CASH FLOW        218113 199253 177703 153274 125770 40980

TAXC                    12345 12468 12593 12719 12846 12975

ENTER SOLVE OPTIONS

? WHAT IF .....WHAT IF コマンド

WHAT IF CASE 1

ENTER STATEMENTS

? SALES = 2950000, PREVIOUS SALES \* 1.028

? SOLVE

ENTER SOLVE OPTIONS

? BEP,NET CASH FLOW

\*\*\*\*\* WHAT IF CASE 1 \*\*\*\*\*

1 WHAT IF STATEMENT PROCESSED

	0	1	2	3	4	5
BEP	2950000	3032600	3117513	3204803	3294538	3386785
NET CASH FLOW	246640	222736	195825	165705	132164	40976

ENTER SOLVE OPTIONS

? GOAL SEEKING.....GOAL SEEKING コマンド

GOAL SEEKING CASE 1

ENTER NAME OF VARIABLE(S) TO BE ADJUSTED TO ACHIEVE PERFORMANCE

? SALES

ENTER 1 COMPUTATIONAL STATEMENT(S) FOR PERFORMANCE

? GROSSMGN = 0

\*\*\*\*\* WHAT IF CASE 1 \*\*\*\*\*

1 WHAT IF STATEMENT PROCESSED

\*\*\*\*\* GOAL SEEKING CASE 1 \*\*\*\*\*

IRR APPROACHING NEGATIVE LIMIT

ERROR IN COMPUTATION OF VARIABLE DEF IN COLUMN 6

	0	1	2	3	4	5
SALES	2127607	2233987	2345687	2462971	2586120	2715426

ENTER SOLVE OPTIONS

? BEP,NET CASH FLOW

	0	1	2	3	4	5
BEP	2127607	2233987	2345687	2462971	2586120	2715426
NET CASH FLOW	-575753	-575876	-576001	-576127	-576254	-630383

ENTER SOLVE OPTIONS

? QUIT

READY

## 【分析例 ANA L 1 7】連結決算子会社の検索とその会社コードのカタログ化

1 >> CFILE NEFA:  
2 >> GET 4 TO 5 ACCITEM=117 & STARTY<91 & DATA(91)=1162:

1990年決算期時点で日経会社コード“1162”を親会社とする子会社の  
レコード(日経親会社コード収録:項目コード=117)を検索  
《記》 ACCITEM (財務項目欄名)、STARTY (収録開始期欄名)

3 >> PERIOD 89 TO 92:  
4 >> DTM:

```
## "STD-DEV" CAN NOT CALCULATE BECAUSE DATA LESS THAN 0
DATA=      1162.0      1162.0      1162.0      1162.0
  1  4232117      TOSHIBA CHEMICAL      親会社日経コード
  2  5213117      TOSHIBA CERAMICS      親会社日経コード
  3  5462117      TOSHIBA STEEL TUBE      親会社日経コード

-----|-----|-----|-----|
      | 4232117 | 5213117 | 5462117 | |
-----|-----|-----|-----|
89.01 |      1162 |      1162 |      1162 | |
90.01 |      1162 |      1162 |      1162 | |
91.01 |      1162 |      1162 |      1162 | |
92.01 |      1162 |      1162 |      1162 | |
-----|-----|-----|-----|
TOTAL |      4648 |      4648 |      4648 | |
MEAN  |      1162 |      1162 |      1162 | |
STD.DEV. |          0 |          0 |          0 | |
```

5 >> PC %%RENO3A=BKEY FORMAT='A4,','':

検索されたレコード全てについてペースキー (BKEY) の頭4桁(会社コード・証券)  
をコンマで区切ったコード・リストを“%%RENO3A”の名前でカタログする

6 >> INFO %%RENO3A:  
1 %%RENO3A 4232,5213,5462,  
7 >> GETR #048 REPEAT %%RENO3A:  
048は、資本金コード  
8 >> DATA:

```
  1  4232048      TOSHIBA CHEMICAL      資本金
89.01 |          | 6000      10172      10172      10172      0

  2  5213048      TOSHIBA CERAMICS      資本金
89.01 |          | 16399     17761     17852     17920      0

  3  5462048      TOSHIBA STEEL TUBE      資本金
89.01 |          | 1260      1260      1260      1260      0
```

【分析例 ANAL18】親会社に対する子会社の構成費の計算出力  
 《使用カタログ, 変数について》

##RENO3I(項目名リスト), ##RENO3K(会社名リスト)  
 @CONO(構成費出力会社: 検索順序番号指定)

```
1 >> CFILE NEFA:
2 >> PERIOD 88 TO 92:
3 >> GETR ##RENO3I REPEAT ##RENO3K:
```

```
4 >> LET @CONO=1;
5 >> @ANS=@COUNT;
6 >> @ANS1=@ANS+1;
7 >> @NOK=NOWD(##RENO3K);
8 >> @NOI=NOWD(##RENO3I);
9 >> @KS=@NOI*@CONO;
10 >> @KS1=@KS+1;
11 >> @KT=@KS+@NOI;
```

```
12 >> DO @KM=1 TO @NOI:
13 >> LET @KS=@KS+1;
14 >> @ANS=@ANS+1;
15 >> #(@ANS)=#(@KS)/#(@KM):
16 >> END:
```

```
17 >> DTM #(1) TO #(@NOI):
```

1 6502055	TOSHIBA	売上高営業収益
2 6502056	TOSHIBA	売上原価営業原価
3 6502058	TOSHIBA	販売費および一般管理費
4 6502070	TOSHIBA	経常利益
5 6502086	TOSHIBA	当期利益

	6502055	6502056	6502058	6502070	6502086
88.01	3572435	2572914	889437	125474	60711
89.01	3800857	2584153	968684	234829	119402
90.01	4251953	2863961	1072059	269736	131836
91.01	4695394	3178509	1254782	258853	120852
92.01	4722383	3310572	1293351	114857	39487
TOTAL	21043008	14510109	5478313	1003749	472288
MEAN	4208601	2902021	1095662	200750	94458
STD.DEV.	463446	301385	157237	66844	37085

## 【分析例 ANAL19】親会社に対する子会社の構成費の計算出力 (ANAL18 続き)

```

1 >> DTM #(BKS1) TO #(BKS), #(BANS1) TO #(BANS):
1 4232055 TOSHIBA CHEMICAL 売上高営業収益
2 4232056 TOSHIBA CHEMICAL 売上原価営業原価
3 4232058 TOSHIBA CHEMICAL 販売費および一般管理費
4 4232070 TOSHIBA CHEMICAL 経常利益
5 4232086 TOSHIBA CHEMICAL 当期利益

```

	4232055	4232056	4232058	4232070	4232086
88.01	39611	32299	4032	2705	1235
89.01	42095	34465	4486	3460	1374
90.01	47509	38668	4896	3852	1825
91.01	51289	40965	5620	5213	2690
92.01	46433	37859	5598	3288	1774
TOTAL	226937	184256	24632	18518	9098
MEAN	45387	36851	4926	3704	1820
STD.DEV.	4114	3087	621	840	482

```

6 #(BANS)
7 #(BANS)
8 #(BANS)
9 #(BANS)
10 #(BANS)

```

	#(BANS)	#(BANS)	#(BANS)	#(BANS)	#(BANS)
88.01	0.01109	0.01255	0.00453	0.02156	0.02034
89.01	0.01108	0.01334	0.00463	0.01473	0.01318
90.01	0.01117	0.01350	0.00457	0.01428	0.01384
91.01	0.01092	0.01289	0.00448	0.02014	0.02226
92.01	0.00983	0.01144	0.00433	0.02863	0.04493
TOTAL	0.05409	0.06372	0.02254	0.09934	0.11455
MEAN	0.01082	0.01274	0.00451	0.01987	0.02291
STD.DEV.	0.00050	0.00073	0.00010	0.00524	0.01156

```
2 >> INFO W#RENO3:
```

```

1 W#RENO3A 4232.5213.5462.
2 W#RENO3B TC055 TC086
3 W#RENO3I #055 #056 #058 #070 #086
4 W#RENO3IB #311000 #312000 #317000 #329000 #390000
5 W#RENO3K 6502.4232.5213.5462

```

```
3 >> D SCAL BNOI, BNOK, BCOUNT:
```

```

      BNOI          5.000      5.0000000E+00
      BNOK          4.000      4.0000000E+00
      BCOUNT      25.000      2.5000000E+01

```

【分析例 ANAL 20】単独決算／連結決算対応表出力  
 《使用カタログ、変数について》

\*\*\*RENO3I(項目名リスト)

```

1 >> FILE KN1A:
2 >> CFILE NEFA:
3 >> PERIOD 88 TO 92:
4 >> GETR ***RENO3I REPEAT 6502:
5 >> GET KN1A.6502311000 KN1A.6502312000 KN1A.6502317000 ADD:
6 >>     KN1A.6502329000 KN1A.650239000 ADD:
    カレント・ファイル以外を検索する場合はコードの頭にDB名を指定する

7 >> LET @ANS=@COUNT:
8 >> @NOI=@NOWD(***RENO3I):
9 >> @KS=@NOI:

10 >> DO @KM=1 TO @NOI:
11 >>     LET @KS=@KS+1:
12 >>         @ANS=@ANS+1:
13 >>         #(@ANS)=#(@KS)/#(@KM):
14 >>     END:
    
```

連結決算・売上高等の表

15 >> DTM # (1) TO # (5):				
1 6502055	TOSHIBA	売上高営業収益		
2 6502056	TOSHIBA	売上原価営業原価		
3 6502058	TOSHIBA	販売費および一般管理費		
4 6502070	TOSHIBA	経常利益		
5 6502086	TOSHIBA	当期利益		

	6502055	6502056	6502058	6502070	6502086
88.01	3572435	2572914	889437	125474	60711
89.01	3800857	2584153	968684	234829	119402
90.01	4251953	2863961	1072059	269736	131836
91.01	4693394	3178509	1254782	258853	120852
92.01	4722383	3310572	1293351	114857	39487
TOTAL	21043008	14510109	5478313	1003749	472288
MEAN	4208601	2902021	1095662	200750	94458
STD.DEV.	463446	301385	157237	66844	37085



〈付録 3〉 興銀財務データ項目 (DB名: 1部 KN1A, 2部 KN2A)

コード	項目名	コード	項目名
110000	流動資産	123025	" (うち関係会社社債)
111000	当座資産	123040	長期貸付金
111010	現預金	123045	" (うち株主役員従業員長期貸付金)
111020	受取手形	123050	関係会社長期貸付金
111030	関係会社受取手形	123070	その他の投資資産
111040	売掛金	123073	" (うち長期前払費用)
111050	関係会社売掛金	123075	" (うち更正債権等)
111060	有価証券	123080	その他の関係会社投資資産
111070	△貸倒引当金等	123085	" (うち関係会社出資金)
112000	棚卸資産	123990	△貸倒引当金等
112010	製品	130000	繰延資産
112020	商品	130040	開発費試験研究費
112030	半製品	130060	その他の繰延資産
112040	仕掛品	190000	資産合計
112050	原材料	210000	負債
112060	貯蔵品	211000	流動負債
113000	その他の流動資産	211010	支払手形
113010	前渡金	211020	関係会社支払手形
113020	関係会社前渡金	211030	設備関係支払手形
113030	前払費用	211035	" (うち関係会社設備関係支払手形)
113040	未収入金	211040	買掛金
113050	関係会社未収入金	211050	関係会社買掛金
113060	未収収益	211060	短期借入金
113070	関係会社未収収益	211065	" (うち関係会社短期借入金)
113080	短期貸付金	211080	1年以内返済長期借入金
113090	関係会社短期貸付金	211085	" (うち関係会社1年以内返済分)
113110	棚卸資産その他	211100	1年以内償還社債
113111	" (うち自己株式)	211120	未払金
113112	" (うち親会社株式)	211123	" (うち関係会社未払金)
113113	" (うち短期金融資産)	211125	" (うち未払税金)
113118	" (うちその他の流動資産)	211130	設備関係未払金
113119	" (うちその他の関係会社流動資産)	211140	未払費用
120000	固定資産	211145	" (うち関係会社未払費用)
121000	有形固定資産	211150	前受金
121010	建物	211160	預り金
121020	構築物	211165	" (うち関係会社預り金)
121030	機械装置	211170	株主・役員・従業員預り金
121050	船舶車両運搬具	211180	前受収益
121060	工具器具備品	211190	割賦販売未実現利益
121070	その他の有形固定資産	211200	その他流動負債
121080	土地	211205	" (うちその他の関係会社流動負債)
121090	建設仮勘定	211210	負債性引当金
122000	無形固定資産	211213	" (うち賞与引当金)
123000	投資その他の資産	211215	" (うち返品調整引当金)
123010	投資有価証券	211217	" (うちその他の負債性引当金)
123020	関係会社投資有価証券	211220	未払法人税等
123023	" (うち関係会社株式)	211223	" (うち未払法人税)

## 興銀財務データ項目 (DB名： 1部 KN1A, 2部 KN2A)

コード	項 目 名	コード	項 目 名
211225	" (うち未払事業税)	312000	売上原価
212000	固定負債	312010	期首製品商品棚卸高
212010	社債	312020	当期製品製造原価
212013	" (うち転換社債)	312030	当期商品仕入高
212030	長期借入金	312035	" (うち関係会社仕入高)
212035	" (うち関係会社長期借入金)	312040	期末製品商品棚卸高
212050	長期支払手形	312050	物品税等
212053	" (うち関係会社長期支払手形)	312060	他勘定振替高・原価差額
212055	" (うち設備関係長期支払手形)	312070	製品商品原材料評価損
212060	長期未払金	312080	売上原価その他
212063	" (うち関係会社長期未払金)	315000	割賦販売利益・返品調整額
212065	" (うち設備関係長期未払金)	316000	売上総損益
212070	その他固定負債	317000	販売費および一般管理費
212075	" (うちその他関係会社固定負債)	317010	販売手数料
212080	負債性引当金	317020	荷造運搬費
212083	" (うち退職給与引当金)	317023	" (うち運送費)
212085	" (うちその他の負債性引当金)	317025	" (うち荷造保管料)
213000	特定引当金(旧商法)	317030	広告宣伝費
213010	為替変動準備金	317040	その他販売費
213090	その他の特定引当金	317050	特許使用料
220000	資本	317060	貸倒引当金繰入額および貸倒損失
221000	資本金	317070	役員報酬
222000	新株式払込金	317080	従業員給料手当
223010	資本準備金	317083	" (うち従業員給料)
223030	その他の資本剰余金	317085	" (うち従業員賞与)
224010	利益準備金	317087	" (うち賞与引当金繰入額)
224020	任意積立金	317090	退職給与引当金繰入額および退職金
224021	" (うち中間配当積立金)	317093	" (うち退職給与引当金繰入額)
224022	" (うち価格変動準備金)	317095	" (うち退職金)
224023	" (うち特別償却準備金)	317097	" (うち退職年金掛金)
224024	" (うち海外投資等損失準備金)	317100	福利厚生費
224025	" (うち海外市場開拓準備金)	317110	旅費通信費
224026	" (うち公舎防止準備金)	317113	" (うち旅費交通費)
224027	" (うち圧縮記帳引当金)	317115	" (うち通信費)
224030	当期末処分利益金	317120	交際費
225030	" (うち社外分配金)	317130	減価償却費
290000	負債・資本合計	317140	貸借料
291000	割引手形	317150	租税公課
292000	関係会社割引手形	317155	" (うち事業税)
294000	譲渡手形	317160	物品税
294005	" (うち関係会社譲渡手形)	317170	試験研究費
295000	保証債務	317180	販売費および一般管理費その他
311000	売上高	317183	" (うち消耗品・事務用品費)
311005	" (うち関係会社売上高)	317185	" (うち動力・用水・光熱費)
311010	製品売上高	317187	" (うちその他の一般管理費)
311020	商品売上高	319000	営業損益
311030	その他の収益	321000	営業外収益

## 興銀財務データ項目 (DB名： 1部 KN1A, 2部 KN2A)

コード	項目名	コード	項目名
321010	受取利息・配当金	332000	特別損失及び特定引当金繰入額
321013	(うち受取利息)	332010	固定資産圧縮・処分損
321015	(うち有価証券利息)	332015	(うち固定資産圧縮損)
321017	(うち受取配当金)	332020	有価証券売却損
321020	関係会社受取利息・配当金	332030	貸倒損失・引当金繰入額
321023	(うち関係会社受取利息)	332040	退職給与引当金繰入額・退職金
321025	(うち関係会社受取配当金)	332043	(うち退職給与引当金繰入額)
321040	資産処分評価益	332045	(うち特別退職金)
321045	(うち有価証券売却益)	332050	減価償却費
321080	その他営業外収益	332060	その他の引当金繰入額
321081	(うち仕入割引)	332070	前期損益修正損
321083	(使用せず)	332080	その他の特別損失
321085	(うち為替差益)	332083	(うち過去勤務債務費用)
321088	(うちその他の関係会社営業外収益)	332085	(うち為替差損)
321089	(うちその他の営業外収益)	332090	特定引当金(旧商法)繰入額
322000	営業外費用	332093	(うち為替変動準備金繰入額)
322010	支払利息・割引料	332095	(うちその他の引当金繰入額)
322013	(うち支払利息・割引料)	333000	税引前当期損益
322015	(うち関係会社支払利息・割引料)	334000	法人税等
322017	(うち社債利息)	334001	うち過年度法人税等追徴額)
322020	社債発行差金償却及び費用	334002	(うち過年度法人税等還付額)
322030	租税公課	390000	当期損益
322050	資産処分評価損	394010	前期繰越損益
322055	(うち有価証券売却評価損)	394020	中間配当積立金取崩額
322070	貸倒引当金繰入額	394030	その他の任意積立金取崩額
322090	減価償却費	394040	中間配当額
322100	その他営業外費用	394050	中間配当に伴う利益準備金積立額
322101	(うち売上割引)	395000	当期末処分利益金
322103	(うち開発費)	395100	任意積立金取崩額
322105	(うち株式発行費)	395200	利益準備金・資本準備金取崩額
322107	(うち為替差損)	396000	利益処分額
322109	(うちその他の営業外費用)	396100	利益準備金積立額
329000	経常損益	396200	配当金
331000	特別利益及び特定引当金取崩額	396300	役員賞与金
331010	固定資産売却益	396400	任意積立金
331020	有価証券売却益	396410	(うち中間配当積立金)
331030	貸倒引当金戻入額	397000	次期繰越利益(欠損)金
331040	退職給与引当金戻入額	411000	原材料費
331050	その他の引当金戻入額	412000	労務費
331060	前期損益修正益	413000	経費
331063	(うち過年度償却済債権取立益)	413010	(うち外注加工費)
331065	(うちその他前期損益修正益)	413020	(うち動力燃料費)
331070	その他の特別利益	413030	(うち人件費)
331075	(うち為替差益)	413070	(うち減価償却費)
331080	特定引当金(旧商法)取崩額	413100	(うち賃借料)
331083	(うち為替変動準備金戻入額)	413110	(うち租税公課)
331085	(うちその他の引当金戻入額)	414000	(その他)

## 興銀財務データ項目 (DB名: 1部 KN1A, 2部 KN2A)

コード	項目名	コード	項目名
415000	他勘定振替高	507090	(うち機械装置)
419000	当期製造総費用	507110	(うち船舶車輛運搬具)
421000	仕掛品・半製品期首棚卸高	507120	"(うち工具器具備品)
422000	仕掛品・半製品期末棚卸高	507130	"(うちその他の有形固定資産)
423000	他勘定振替高・原価差額	507140	無形固定資産減価償却額
490000	当期製品製造原価	507150	投資その他の資産減価償却額
500010	営業月数	507160	繰延資産減価償却額
501010	額面株式の券面額	507170	減価償却累計額
501020	発行済株式総数	507180	有形固定資産減価償却累計額
501030	株式総数	507190	"(うち建物)
501040	政府地方公共団体持株数	507200	"(うち構築物)
501050	金融機関持株数	507210	"(うち機械装置)
501060	会社持株数	507230	(うち船舶車両運搬具)
501070	その他の法人持株数	507240	"(うち工具器具備品)
501080	外国法人等持株数	507250	"(うちその他の有形固定資産)
501090	"(うち個人持株数)	507260	無形固定資産減価償却累計額
501100	個人その他持株数	507270	投資その他の資産減価償却累計額
502010	期末従業員数	507280	繰延資産減価償却累計額
502030	期末従業員の平均年齢	507290	減価償却の方法
502040	期末従業員の平均給与月額	507300	償却範囲額に対する過不足額(当期)
503010	監査意見	507310	有形固定資産償却範囲額に対する過不足
505010	棚卸資産評価の基準及び方法(商品)	507320	無形固定資産償却範囲額に対する過不足
505020	"(製品)	507330	償却範囲額に対する過不足額(累計額)
505030	"(半製品)	507340	有形固定資産償却範囲額に対する過不足
505040	"(仕掛品)	507350	無形固定資産償却範囲額に対する過不足
505050	"(原材料)	507410	有形固定資産当期増加額
505060	"(貯蔵品)	507420	建設仮勘定当期減少額
506000	有価証券の評価基準	508010	社債当期発行額
506010	一時的所有(株式数)	508020	社債当期償還額
506020	"(株式 取得価額)	508030	社債未償還残高
506030	"(株式 貸借対照評価額)	508110	転換社債当期発行額
506040	"(公社債国債地方債 取得価額)	508120	転換社債当期償還額又は転換額
506050	"(公社債国債地方債 貸借対照表価)	508130	転換社債未償還残高
506060	"(その他の有価証券 取得価額)	508210	新株引受権付社債当期発行額
506070	"(その他の有価証券 貸借対照表価)	508220	新株引受権付社債当期償還額
506110	投資(株式数)	508230	新株引受権付社債未償還残高
506120	"(株式取得価額)	508500	長期借入金当期借入額
506130	"(株式貸借対照表価額)	508510	長期借入金当期返済額
506140	"(公社債国債地方債取得価額)	508600	期中有償増資額
506150	"(公社債国債地方債貸借対照表価額)	508610	期中有償増資による資本準備金増加額
506160	"(その他の有価証券取得価額)	508620	"(うち期中転換社債転換による増資)
506170	"(その他の有価証券貸借対照表価額)	508630	"(うち期中転換社債転換による資本)
507040	減価償却額	508640	期中無償増資額
507050	(うち特別減価償却額)	508650	"(うち資本準備金取崩額)
507060	"(有形固定資産減価償却額)	508660	期中減資額
507070	"(うち建物)	509000	外貨建長期金銭債権
507080	(うち構築物)	509010	"(同上決算日レート換算額)

## 興銀財務データ項目（DB名： 1部 KN1A, 2部 KN2A）

コード	項目名	コード	項目名
509020	外貨建長期金銭債務	550010	1株当たり現金配当
509030	"（同上決算日レート換算額）	550013	"（うち記念配当）
509040	外貨建投資有価証券	550015	"（うち特別配当）
509050	"（うち外貨建関係会社株式）	550020	100株当たり株式配当
509060	外貨建一時所有の有価証券	550030	1株当たり中間配当
509070	外貨建社債未償還残高	550040	1株当たり当期損益
509080	"（うち外貨建転換社債）	560010	当期受注高
509200	担保差入資産	560020	受注残高
509400	貸倒引当金, 当期増加額	560030	所有土地面積
509410	" , 当期目的使用額	560040	所有土地簿価
509420	" , 当期その他使用額	560050	企業担保
509430	負債性引当金, 当期増加額	560070	圧縮配債直接控除額
509440	" , 当期目的使用額	560080	合併分離フラッグ
509450	" , 当期その他使用額	610000	決算期
509460	特定引当金, 当期増加額	610100	決算月数
509470	" , 当期目的使用額	611000	売上高
509480	" , 当期その他使用額	611100	経常損益
510010	現金	611200	当期損益
510020	預金	612000	1株当たり配当金合計
510030	"（うち当座普通通知預金）	612100	"（うち普通配当）
510040	"（うち譲渡性預金）	612200	"（うち記念配当）
510050	"（うち定期預金）	612300	"（うち特別配当）
510060	"（うち外貨預金）	612400	配当区分
510070	"（うち金銭の信託）	620000	決算期
510080	"（うちその他の預貯金）	620100	決算月数
511010	売上の内訳 項目名（1）	621000	売上高
511020	" 金額（1）	621100	経常損益
511030	" 項目名（2）	621200	当期損益
511040	" 金額（2）	622000	1株当たり配当金合計
511050	" 項目名（3）	622100	"（うち普通配当）
511060	" 金額（3）	622200	"（うち記念配当）
511070	" 項目名（4）	622300	"（うち特別配当）
511080	" 金額（4）	622400	配当区分
511090	" 項目名（5）	700100	鉄軌道業（営業収入）
511100	" 金額（5）	700110	"（営業費用）
511110	" 項目名（6）	700120	"（運送費）
511120	" 金額（6）	700130	"（ "（うち人件費））
511130	" 項目名（7）	700140	"（一般管理費）
511140	" 金額（7）	700150	"（ "（うち人件費））
511150	" 項目名（8）	700160	"（租税公課）
511160	" 金額（8）	700170	"（減価償却費）
511170	" 項目名（9）	700180	"（営業損益）
511180	" 金額（9）	700200	旅客自動車業（営業収入）
511190	" 項目名（10）	700210	"（営業費用）
511200	" 金額（10）	700220	"（運送費）
511220	金額合計	700230	"（ "（うち人件費））
511230	売上高中の輸出高	700240	"（一般管理費）

## 興銀財務データ項目 (DB名： 1部 KN1A, 2部 KN2A)

コード	項目名	コード	項目名
700250	" ( " (うち人件費)	701230	" (貨車走行キロ数)
700260	" (営業損益)	701240	" (在籍車両数)
700300	貨物自動車業 (営業収入)	701250	" ( " , うち客車数)
700310	" (営業費用)	701260	" (乗車効率)
700320	" (運送費)	701270	" (期末従業員数)
700330	" ( " (うち人件費)	702100	旅客自動車業 (営業日数)
700340	" (一般管理費)	702110	" (乗合営業キロ数)
700350	" ( " (うち人件費)	702120	" (乗車人員)
700360	" (営業損益)	702130	" ( " , うち乗合)
700400	通運業 (営業収入)	702140	" ( " , うち貸切)
700410	" (営業費用)	702150	" (旅客収入)
700420	" (運送費)	702160	" ( " , うち乗合)
700430	" ( " (うち人件費)	702170	" ( " , うち貸切)
700440	" (一般管理費)	702180	" (運輸雑収)
700450	" ( " (うち人件費)	702190	" (その他の収入)
700460	" (営業損益)	702200	" (乗合・貸切バス業営業収入)
700500	兼業 (営業収入)	702210	" (乗合・貸切バス走行キロ数)
700510	" (不動産営業収入)	702220	" ( " , うち乗合)
700520	" (不動産分譲収入)	702230	" ( " , うち貸切)
700530	" (不動産賃貸収入)	702240	" (在籍車両数)
700540	" (その他事業収入)	702250	" ( " , うち乗合)
700550	" (営業費用)	702260	" ( " , うち貸切)
700560	" (商品等仕込原価)	702270	" (乗合乗車効率)
700570	" (不動産分譲業)	702280	" (期末従業員数)
700580	" (その他事業)	703100	通運業・貨物自動車業 (貨自業輸送トン
700590	" (人件費)	703110	" (通運業取扱トン数)
700600	" (経費)	703120	" (貨物自動車業営業収入)
700610	" (租税公課)	703130	" (通運業営業収入)
700620	" (減価償却費)	703140	" (在籍車両数)
700630	" (営業損益)	703150	" (期末従業員数)
700700	「その他一般管理費等」	703160	" (路線期末営業キロ数)
700800	合計 (営業収入)	703170	" (路線運送トン数)
700810	" (営業費用)	703180	" (路線運送収入)
700820	" (営業損益)	704100	鉄軌道業固定資産
701100	鉄軌道業 (営業日数)	704110	旅客自動車業固定資産
701110	" (営業キロ数)	704120	貨物自動車業固定資産
701120	" (乗車人員)	704130	通運業固定資産
701130	" ( " , うち定期)	704140	兼業固定資産
701140	" ( " , うち定期外)	704150	" (うち不動産業固定資産)
701150	" (旅客収入)	704160	各事業関連固定資産
701160	" ( " , うち定期)	704170	建設仮勘定
701170	" ( " , うち定期外)	704180	" (うち鉄軌道業)
701180	" ( " , うち手小荷物)	704190	" (うち自動車業)
701190	" (貨物収入)	704200	" (うちその他)
701200	" (運輸雑収)	704210	投資その他の資産
701210	" (鉄軌道業営業収入)	704220	固定資産計
701220	" (客車走行キロ数)		

〈付録 4〉 COMPUSTAT財務データ  
項目 (DB名: CSSA)

コード	項目名	コード	項目名
001	CASH AND SHORT-TERM INVESTMENTS	046	RESEARCH AND DEVELOPMENT EXPENSE
002	RECEIVABLES	047	RENTAL EXPENSE
003	INVENTORIES	048	E.I. & D.O.
004	CURRENT ASSETS(TOTAL)	049	MINORITY INTEREST(INCOME ACCOUNT)
005	CURRENT LIABILITIES(TOTAL)	050	DEFERRED TAXES(INCOME ACCOUNT)
006	ASSETS/LIABILITIES AND SHAREHOLDERS'EQUITY	051	INVESTMENT TAX CREDIT(INCOME ACCOUNT)
007	PLANT-GROSS	052	TAX LOSS CARRY FORWARD
008	PLANT-NET	053	EARNING PER SHARE(PRIMARY)-INCLUDING E.I. & D.
009	LONG-TERM DEBT(TOTAL)	054	COMMON SHARES USED TO CALCULATE PRIMARY
010	PREFERRED STOCK AT LIQUIDATING VALUE	055	UNCONSOLIDATED SUBSIDIARIES-EQUITY IN EARNINGS
011	COMMON EQUITY(TANGIBLE)	056	PREFERRED STOCK AT REDEMPTION VALUE
012	SALES-NET	057	EARNING PER SHARE-EXCLUDING E.I. & D.O.
013	OPERATING INCOME BEFORE DEPRECIATION	058	EARNING PER SHARE(PRIMARY)-EXCLUDING E.I. & D.O.
014	DEPRECIATION AND AMORTIZATION	059	INVENTORY VALUATION METHOD
015	INTEREST EXPENSE	060	COMMON EQUITY(AS REPORTED)
016	INCOME TAXES(TOTAL)	061	NONOPERATING INCOME/EXPENSE
017	SPECIAL ITEMS	062	INTEREST INCOME
018	INCOME BEFORE E.I. & D.O.	063	INCOME TAXES-FEDERAL
019	PREFERRED DIVIDENDS	064	INCOME TAXES-FOREIGN
020	AVAILABLE FOR COMMON AFTER ADJUSTMENTS FOR COM	065	AMORTIZATION OF INTANGIBLES
021	COMMON DIVIDENDS	066	DISCONTINUED OPERATIONS
022	PRICE-HIGH	067	RECEIVABLES(ESTIMATED DOUBTFUL)
023	PRICE-LOW	068	CURRENT ASSETS(OTHER)
024	PRICE-CLOSE	069	ASSETS(OTHER)
025	COMMON SHARES OUTSTANDING	070	ACCOUNT PAYABLE
026	DIVIDENDS PER SHARE	071	INCOME TAXES PAYABLE
027	ADJUSTMENT FACTOR(CUMULATIVE)	072	CURRENT LIABILITIES(OTHER)
028	COMMON SHARES TRADED	073	CONSTRUCTION IN PROGRESS
029	EMPLOYEES	074	DEFERRED TAXES(BALANCE SHEET)
030	CAPITAL EXPENDITURES	075	LIABILITIES(OTHER)
031	INVESTMENTS AND ADVANCES-EQUITY METHOD	076	RAW MATERIALS
032	INVESTMENTS AND ADVANCES-OTHER	077	WORK IN PROCESS
033	INTANGIBLES	078	FINISHED GOODS
034	DEBT IN CURRENT LIABILITIES	079	DEBT(CONVERTIBLE)
035	DEFERRED TAXES AND INVESTMENT TAX CREDIT	080	DEBT(SUBORDINATED)
036	RETAINED EARNINGS	081	DEBT(NOTES)
037	INVESTED CAPITAL(TOTAL)	082	DEBT(DEVENTURES)
038	MINORITY INTEREST(BALANCE SHEET)	083	DEBT(OTHER LONG-TERM)
039	CONVERTIBLE DEBT AND PREFERRED STOCK	084	DEBT(CAPITALIZED LEASE OBLIGATIONS)
040	COMMON SHARES RESERVED FOR CONVERSION	085	COMMON STOCK
041	COST OF GOODS SOLD	086	TREASURY STOCK(TOTAL DOLLAR AMOUNT)
042	LABOR AND RELATED EXPENSE	087	TREASURY STOCK(NUMBER OF COMMON SHARES)
043	PENSION AND RETIREMENT EXPENSE	088	PRESENT VALUE OF NON-CAPITALIZED LEASES
044	DEBT DUE IN ONE YEAR	089	UNFUNDED PENSION COSTS-VESTED BENEFITS
045	ADVERTISING EXPENSE	090	UNFUNDED PENSION COSTS-PAST OR PRIOR SERVICE

# COMPUSTAT財務データ 項目 (DB名: CSSA)

コード	項目名	コード	項目名
091	DEBT MATURING IN THE SECOND YEAR	134	INTEREST EXPENSE(RESTATED)
092	DEBT MATURING IN THE THIRD YEAR	135	INCOME TAXES(TOTAL)-(RESTATED)
093	DEBT MATURING IN THE FOURTH YEAR	136	E.I. & D.O.(RESTATED)
094	DEBT MATURING IN THE FIFTH YEAR	137	EARNING PER SHARE-INCLUDING E.I. & D.O.
095	MINIMUM RENTAL COMMITMENTS IN FIVE YEARS	138	COMMON SHARES USED TO CALCULATE PRIMARY
096	MINIMUM RENTAL COMMITMENT IN FIRST YEAR	139	EARNING PER SHARE-EXCLUDING E.I. & D.O.
097	RETAINED EARNING(UNRESTRICTED)	140	EARNING PER SHARE-INCLUDING E.I. & D.O.
098	ORDER BACKLOG	141	PLANT-NET(RESTATED)
099	RETAINED EARNING(RESTATEMENT)	142	LONG-TERM DEBT(TOTAL)-(RESTATED)
100	COMMON SHAREHOLDERS	143	RETAINED EARNING(RESTATED)
101	INTEREST EXPENSE ON LONG-TERM DEBT	144	STOCKHOLDERS'EQUITY(RESTATED)
102	EXCISE TAXES	145	CAPITAL EXPENDITURES(RESTATED)
103	DEPRECIATION EXPENSE(SCHEDULE VI)	146	EMPLOYEERS(RESTATED)
104	BORROWING-AVERAGE SHORT-TERM	147	INTEREST CAPITALIZED
105	INTEREST RATE	148	LONG-TERM DEBT TIED TO PRIME
106	UNREMITTED EARNING OF UNCONSOLIDATED SUBSIDIAR	149	AUDITOR/AUDITOR'S OPINION
107	SALE OF PROPERTY,PLANT,AND EQUIPMENT	150	FOREIGN CURRENCY ADJUSTMENT
108	SALE OF COMMON AND PREFERRED STOCK	151	TRADE RECEIVABLES
109	SALE OF INVESTMENTS	152	DEFERRED CHARGES
110	TOTAL FUNDS FROM OPERATIONS	153	ACCRUED EXPENSES
111	ISSUANCE OF LONG-TERM DEBT	154	DEBT(CONVERTIBLE SUBORDINATED)
112	TOTAL SOURCES OF FUNDS	155	BUILDINGS-NET
113	INCREASE IN INVESTMENTS	156	EQUIPMENT AND MACHINERY-NET
114	REOUCTION OF LONG-TERM DEBT	157	NATURAL RESOURCES-NET
115	PURCHASE OF COMMON AND PREFERRED STOCK	158	LAND
116	TOTAL USES OF FUNDS	159	LEASES-NET
117	SALES	160	PREPAID EXPENSES
118	INCOME BEFORE E.I. & D.O.	161	INCOME TAX REFUND
119	EARNING PER SHARE-INCLUDING E.I. & D.O.	162	CASH
120	ASSETS(TOTAL)(RESTATED)	163	RENTAL INCOME
121	WORKING CAPITAL(RESTATED)	164	MINIMUM RENTAL COMMITMENT IN SECOND YEAR
122	PRETAX INCOME(RESTATED)	165	MINIMUM RENTAL COMMITMENT IN THIRD YEAR
123	INCOME BEFORE E.I. & D.O.	166	MINIMUM RENTAL COMMITMENT IN FOURTH YEAR
124	E.I. & D.O.	167	MINIMUM RENTAL COMMITMENT IN FIFTH YEAR
125	DEPRECIATION AND AMORTIZATION	168	COMPENSATING BALANCE
126	DEFERRED TAXES	169	EARNING PER SHARE-INCLUDING E.I. & D.O.
127	CASH DIVIDENDS	170	PRETAX INCOME
128	CAPITAL EXPENDITURES	171	COMMON SHARES USED TO CALCULATE FULLY DILUTED
129	ACQUISITIONS	172	NET INCOME(LOSS)
130	PREFERRED STOCK-CARRYING VALUE	173	INCOME TAXES-STATE
131	COST OF GOODS SOLD	174	DEPLETION EXPENSE(SCHEDULE VI)
132	SELLING,GENERAL,AND ADMINISTRATIVE EXPENSES	175	REDEEMABLE PREFERRED STOCK
133	DEPRECIATION AND AMORTIZATION		

—Y—

Y名前 .....	34
Y名前 .....	34

—A—

ALGOL 60 .....	5, 12
ASTRO-FOIL .....	38
ATM (Asynchronous Transfer Mode) .....	111

—B—

BNF記法 .....	1
-------------	---

—C—

C++言語の構文 .....	12
C++言語 .....	38
CODASYL .....	66
COMPSTAT .....	69
Concurrent Pascal .....	63

—D—

DBMS .....	1
DCE .....	82
DDX .....	93

—E—

EXSTAT .....	71
Extel社 (英国) 財務データ .....	71

—F—

FDDI .....	109
FEP .....	93
FORTAN の構文 .....	7

—G—

GOAL SEEKING .....	161
--------------------	-----

—I—

IEEE 802 規格 .....	109
IFPS .....	159
INSネット1500 .....	106
INSネット64 .....	106
IOCS .....	90
ISDN .....	106

—L—

LAN .....	107
LISP .....	6, 63

—M—

Mach .....	98
MIPS .....	87
MML .....	104
MPP .....	103
MULTICS .....	90

—N—

N-1ネットワーク .....	93
-----------------	----

—O—

OSI .....	109
-----------	-----

—P—

PASCAL のデータ宣言文 .....	10
PL/I のデータ宣言 .....	11

—Q—

QBE .....	85
-----------	----

QOS (Quality of Service) ..... 111

—S—

S&P社(米国)財務データ ..... 71

SECRETARY ..... 38, 114

SECRETARYの構文 ..... 24

SIMPLAN ..... 159

Simula 67 ..... 6

Smalltalk ..... 6

SQL ..... 78, 85, 126

STEPS ..... 38

—T—

TCP/IP ..... 109

—U—

UNIX ..... 92

—V—

VSAM ..... 84

—W—

WAN ..... 110

WHAT IF ..... 161

—X—

Xウィンドウ ..... 108

—あ—

アプリケーション・プログラム ..... 81

—い—

インターネット ..... 83, 106

隠蔽(information hiding) ..... 53

—う—

ウィンドウ環境 ..... 81

—え—

枝(branch or pointer) ..... 62

—お—

横断面分析 ..... 69

オブジェクト ..... 53

オブジェクト指向 ..... 42

オブジェクト指向プログラミング ..... 55

オペレーティング・システム ..... 90

親(parent) ..... 62

オープン・システム ..... 82

—か—

仮想記憶方式 ..... 90

仮想端末(Virtual Terminal) ..... 108

株式データ ..... 72

関係演算 ..... 101

関係モデル ..... 101

環状リスト(circular-list) ..... 44

—き—

基幹データベース ..... 99

—ぎ—

行列演算式 ..... 30

行列関数名 ..... 31

行列変数 ..... 29

—く—

クライアント/サーバー・システム ..... 82

クラス・キーワード ..... 13

クラス宣言子 ..... 15

—け—	
継承機能.....	54
—げ—	
言語処理系.....	36
現在価値法.....	136
ゲートウェイ.....	110
—こ—	
子 (child).....	62
興銀財務データ.....	69
構造体共用体指定.....	69
構造体共用体指定子.....	21
構文規則.....	1
構文記法.....	1
コッド (Codd, Edgar F.).....	66
コードブック.....	76
—ざ—	
財務関連データ.....	72
財務データベース.....	68
財務分析 (financial analysis).....	133
—じ—	
時系列分析.....	60
—し—	
資本回収係数.....	136
終価計算.....	136
主成分分析.....	146
小行列関数.....	32
正味現在価値法.....	136
—す—	
趨勢財務諸表分析.....	58

スキーマ.....	100
—せ—	
節 (node).....	62
線形リスト (linear-list).....	44
—そ—	
双方向リスト (doubly-linked-list).....	44
添字付行列変数.....	30
ソフトウェア危機.....	119
損益計算書.....	56
損益分岐点分析.....	142
—た—	
貸借対照表.....	56
多重継承.....	54
多重定義 (overload).....	54
多分木.....	46
多変量解析手法.....	146
—ち—	
抽象データ型.....	42, 51
超言語.....	2
—て—	
手続き型.....	36
—で—	
データ.....	41
データアーカイブ.....	65
データベース・スキーマ.....	86
データベースの発展.....	65
データモデル.....	41
データ型.....	4
データ構造.....	41
データ構造 (配列).....	4

	—ど—		—ぶ—
導出クラス.....	38	プログラミング言語.....	36
	—な—		—ぶ—
内部収益率法.....	137	分散システム.....	92
	—に—	分散データベース.....	85
日経財務データ.....	70	散DBサーバ.....	119
二分木.....	44		—ま—
	—ね—	マルチスレッド.....	126
根 (root).....	62	マルチタスク.....	126
ネットワークモデル.....	109		—み—
年金現価係数.....	136	ミドル・ソフトウェア.....	78
	—の—		—め—
ノイマン型コンピュータ.....	130	メタデータ.....	56
	—は—	メタ記号.....	2
葉 (leaf).....	62	メタ変数.....	2
配列.....	37		—も—
	—ひ—	目的言語.....	2
比較財務諸表分析.....	58		—り—
非手続き型言語.....	36	利益金処分計算書.....	56
百分比比率分析.....	131	利子率関数.....	134
比率分析.....	131	リスト構造.....	43
	—ふ—		—る—
副プログラム.....	5	ルータ (router).....	109
	—ぶ—		
ブリッジ (bridge).....	109		

## 研 究 叢 書 (既 刊)

- 
- |      |   |              |       |
|------|---|--------------|-------|
| 第1号  | 生産と分配に対する貿易効果の分析  | 片野 彦二著       | 1961年 |
| 第2号  | 国際貿易と経済発展   | 川田富久雄著       | 1961年 |
| 第3号  | 国際私法の法典化に関する史的研究  | 川上 太郎著       | 1961年 |
| 第4号  | アメリカ経営史   | 井上 忠勝著       | 1961年 |
| 第5号  | 神戸港における港湾荷役経済の研究<br>柴田銀次郎・佐々木誠治・秋山 一郎・山本 泰督共著                                       |              | 1962年 |
| 第6号  | 企業評価論の研究  | 小野 二郎著       | 1963年 |
| 第7号  | 経営費用理論研究  | 小林 哲夫著       | 1964年 |
| 第8号  | 船内労働の実態   | 佐々木誠治著       | 1964年 |
| 第9号  | 船員の雇用制度   | 山本 泰督著       | 1965年 |
| 第10号 | 国際私法条約集   | 川上 太郎著       | 1966年 |
| 第11号 | 地域経済開発と交通に関する理論   | 野村寅三郎著       | 1966年 |
| 第12号 | 国際私法の国際的法典化   | 川上 太郎著       | 1966年 |
| 第13号 | 南北貿易と日本の政策  | 川田富久雄著       | 1966年 |
| 第14号 | インド経済における所得分配構造   | 片野 彦二著       | 1968年 |
| 第15号 | ラテンアメリカ経済統合の理論と現実   | 西向 嘉昭著       | 1969年 |
| 第16号 | 会計情報とEDP監査  | 中野 勲・大矢知浩司共著 | 1972年 |
| 第17号 | 国際収支と資産選択   | 井川 一宏著       | 1974年 |
| 第18号 | 経営計測システムの研究<br>Business & Economic Information Control and Analysis System<br>定道 宏著 |              | 1978年 |
| 第19号 | 日本・オセアニア間の海上輸送とオセアニア主要渾の現況<br>佐々木誠治著  |              | 1978年 |
| 第20号 | 計量経済システムSTEPS-BEICA   | 定道 宏・布上 康夫共著 | 1979年 |
| 第21号 | 海上運賃の経済分析   | 下條 哲司著       | 1979年 |
| 第22号 | 国際法上の船籍論  | 嘉納 孔著        | 1981年 |
| 第23号 | ブラジル経済の高度成長期の研究   | 西島 章次著       | 1981年 |
-

研究叢書（既刊）

- 第24号 資本蓄積過程の分析  
— 理論的枠組とオーストラリア経済への適用 — 下村 和雄著 1983年
- 第25号 会計情報公開論 山地 秀俊著 1983年
- 第26号 企業の国際化をめぐる特殊研究 井上 忠勝・山本 泰督・  
下條 哲司・井川 一宏・山地 秀俊共著 1983年
- 第27号 海運における国家政策と企業行動 海運経済専門委員会著 1984年
- 第28号 オーストラリアの金融システムと金融政策 石垣 健一著 1985年
- 第29号 会計情報公開制度の実証的研究  
— 日米比較を目指して — 山地 秀俊著 1986年
- 第30号 配船の理論的基礎 下條 哲司編著 1986年
- 第31号 仮想電子計算機と計算機言語システム  
— 世界計量経済モデル分析システム — 安田 聖著 1986年
- 第32号 期待効用理論 — 批判的検討 — 伊藤 駒之著 1986年
- 第33号 アメリカ企業経営史研究 井上 忠勝著 1987年
- 第34号 反トラスト政策 — 経済的および法的分析 —  
カール ケインゼン・ドナルド F. ターナー共著  
根岸 哲・橋本 介三共訳 1988年
- 第35号 会計情報システムと人間行動 中野 勲著 1989年
- 第36号 国際金融経済論の新展開  
— 変動為替相場制度を中心として — 井澤 秀記著 1989年
- 第37号 労働市場研究の現代的課題 小西 康生・三木 信一共著 1989年
- 第38号 香港企業会計制度の研究 中野 勲編著 1989年
- 第39号 国際比較統計研究モノグラフ1 能勢 信子編著 1990年
- 第40号 経済発展と環太平洋経済  
西向 嘉昭・石垣 健一・西島 章次・片山 誠一共編著 1991年
- 第41号 労使問題と会計情報公開 山地 秀俊著 1991年
- 第42号 経営財務と会計の諸問題 森 昭夫編著 1992年
- 第43号 国際比較統計研究モノグラフ2 小西 康生著 1993年
- 第44号 アメリカ現代会計成立史論 中野 常男・高須 教夫・山地 秀俊共著 1993年
- 第45号 ネットワーク環境における情報システムの研究 宮崎 耕著 1994年

平成7年12月18日 印刷 (非売品)

平成7年12月22日 発行

神戸大学経済経営研究所講師

著者 民野庄造

神戸市灘区六甲台町2-1

発行所 神戸大学経済経営研究所

印刷 神戸市中央区中山手通7-5-7  
有限会社 興文社

---