

45-E 001

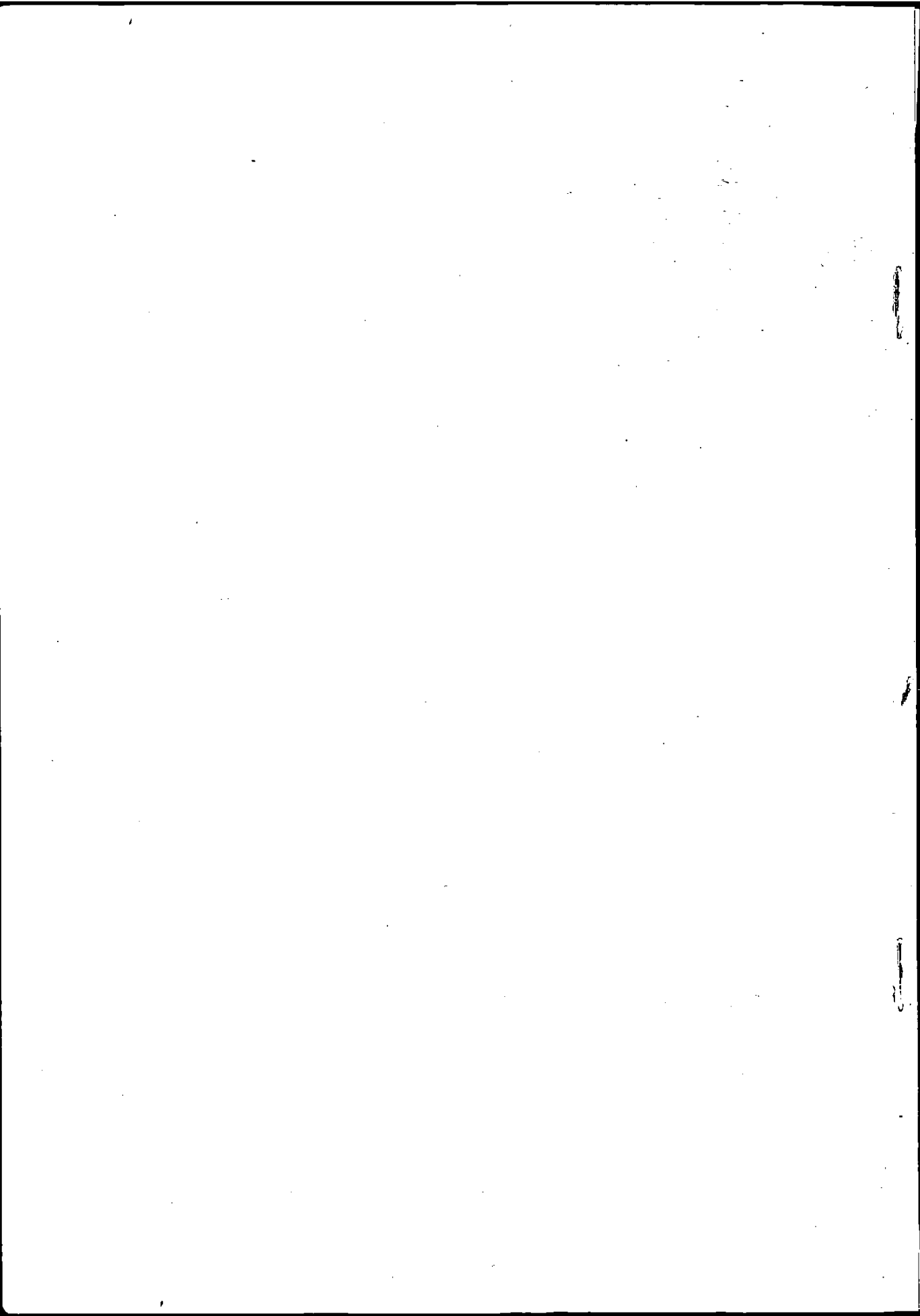
初級情報処理技術者育成指針

昭和46年3月



財団法人 日本情報処理開発センター

本事業は、日本自転車振興会の機械工業振興資金による「昭和
45年度情報処理普及促進に関する補助事業」の一環として実施
したものであります。



序

情報処理および情報処理産業の前途には解決を要する幾多の課題がありますが、とりわけ情報処理技術者の育成確保は最も基本的かつ重要な問題の一つと考えられます。

当財団では情報処理に関する諸問題解決のため各種の事業を実施しておりますが、とくに技術者養成事業の一環として、さきに通商産業省の委託を受けて作成した「上級情報処理技術研修ガイドブック」をはじめとして、情報処理技術者の各レベルに応じた育成指針・学習書などの作成を実施中であります。

この「初級情報処理技術者育成指針」は、その一部として完成したもので、官公庁、民間企業あるいは各種民間養成機関などにおいて、情報処理技術者を志す人々に、その基礎を教育する立場にある教師のための指導要領として活用できるよう、とりまとめたものであります。

すなわち、今後ますます拡大し高度化してゆく情報処理技術の発展とともに、その適用分野もきわめて多岐にわたることが予想されますが、将来これら各専門分野にたずさわる上級技術者に成長してゆくために、初級情報処理技術者として、修得しておくべき共通的基本的事項を明確にし、わが国における初級技術者の位置づけと、これの教育目標、方法、カリキュラム編成などについての指針を示したものであります。

ここに、本育成指針の作成にご尽力ならびにご支援を賜った関係各位に心から感謝の意を表しますとともに、この育成指針が、各方面に利用され、わが国情報処理技術者育成の基礎固めの一助として寄与できますよう念願いたす次第であります。

昭和46年3月

財団法人 日本情報処理開発センター

会長 難波捷吾

初級情報処理技術者育成指針委員会

(敬称略 50音順)

委員長	山 内 二 郎	青山学院大学
委 員	安 西 美 臣	日本電信電話公社
"	今 村 茂 雄	(財)情報処理研修センター
"	魚 木 五 夫	広島商科大学
"	雲 英 道 夫	文 部 省
"	篠 崎 敬	(財)日本情報処理開発センター
"	杉 山 和 男	通商産業省
"	関 重 広	日本電子工学院
"	関 口 修	文 部 省
"	田 島 一 郎	慶応義塾大学
"	西 尾 出	三井情報開発(株)
"	山 本 欣 子	(財)日本情報処理開発センター

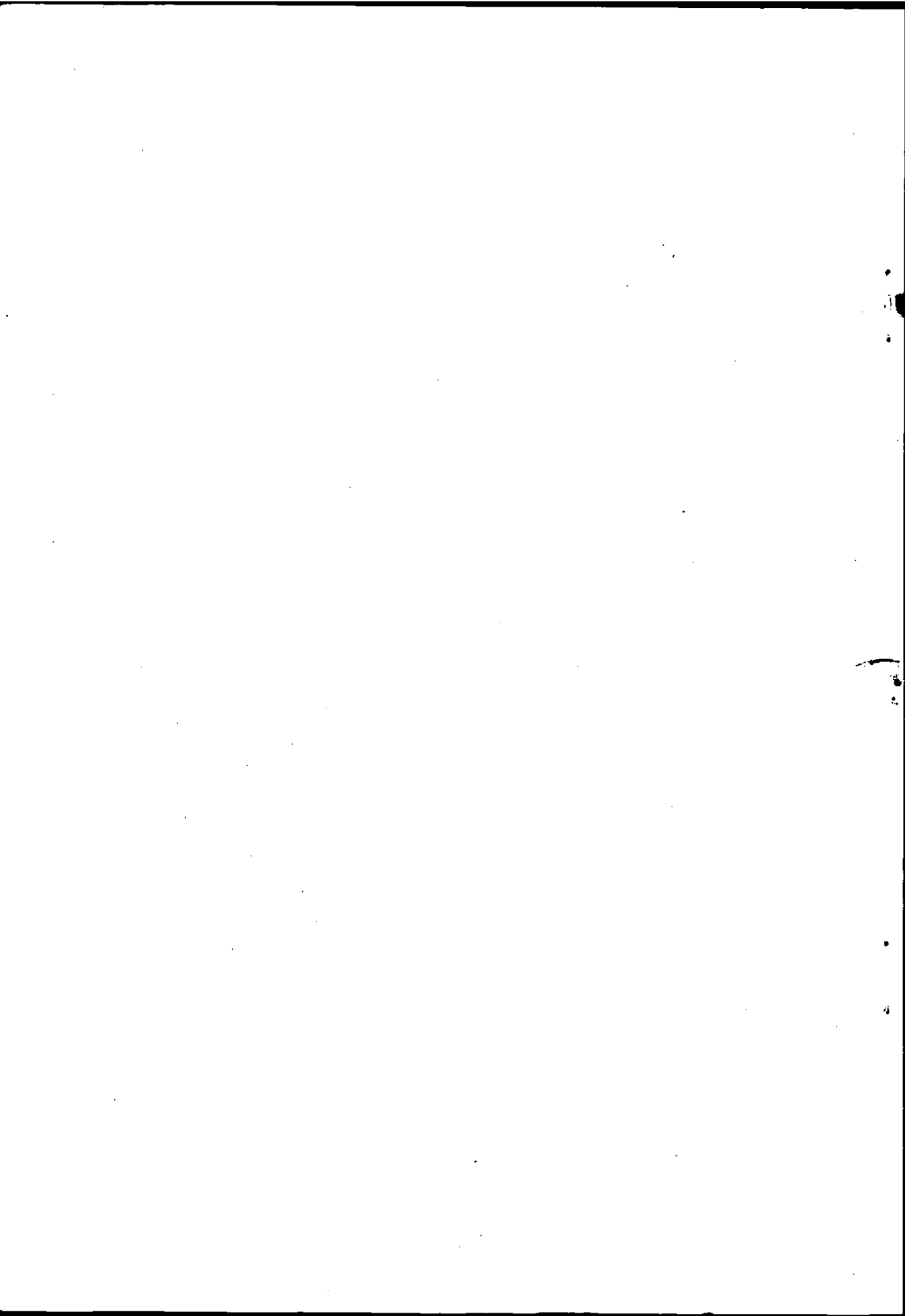
初級情報処理技術者育成指針作成小委員会

(敬称略 50音順)

委員長	魚 木 五 夫	広島商科大学
委 員	安 西 美 臣	日本電信電話公社
"	大日方 真	日本ソフトウェア(株)
"	山 本 欣 子	(財)日本情報処理開発センター

事務局 (財)日本情報処理開発センター

技 術 部 教 育 課



初級情報処理技術者育成指針

総 目 次

第1編 総 論

1. 情報処理技術者の育成	1
2. 初級情報処理技術者育成指針の位置づけ	4
3. カリキュラムの編成	11
4. 学習指導計画と指導法	17

第2編 各 論

基 礎 編

	23
--	----

第1部 データ処理の基礎

	23
--	----

第1章 事務とデータ処理

	24
--	----

第2章 機械によるデータ処理

	34
--	----

第3章 データ収集

	38
--	----

第2部 コンピュータのハードウェア

	47
--	----

第1章 ハードウェアのあらまし

	48
--	----

第2章 入出力装置

	53
--	----

第3章 記憶装置

	60
--	----

第4章 演算・制御装置

	67
--	----

第3部 コンピュータのソフトウェア

	75
--	----

第1章 プログラムのあらまし

	76
--	----

第2章 プログラミング・システム

	84
--	----

第3章 言語のあらまし

	94
--	----

第4部 コンピュータの利用

	103
--	-----

第1章 システム設計

	104
--	-----

第2章 コンピュータの応用

	121
--	-----

実 習 編

第1部 プログラミング (1)

	129
--	-----

FORTRAN

	129
--	-----

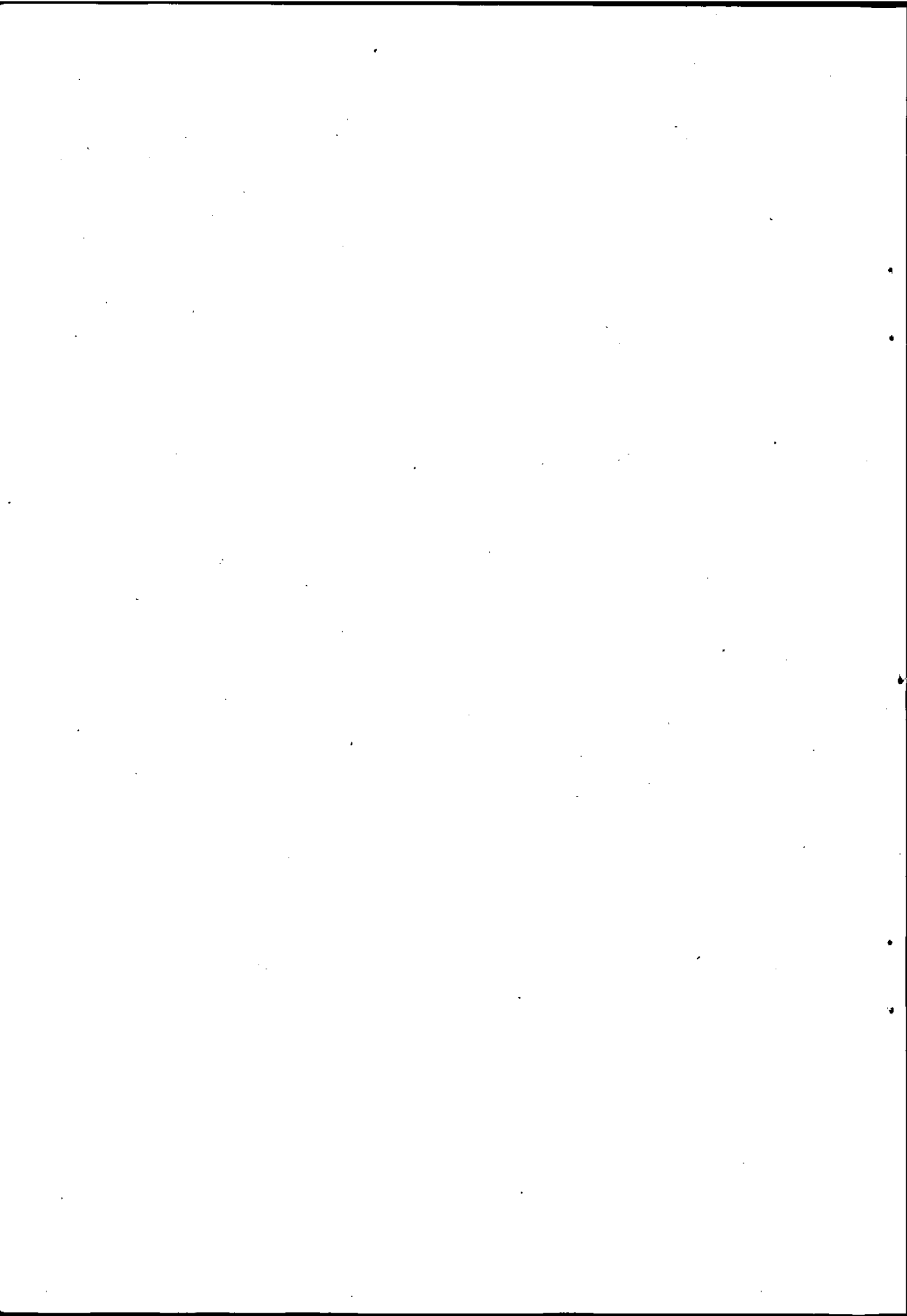
第2部 プログラミング (2)	165
COBOL	165
第3部 データ準備	201
第4部 コンピュータの運用	215
第3編 付 録	
1. 情報処理技術者試験の概要	231
2. 情報処理関係の標準・規格	249
3. 情報処理用語	381
4. 教育施設・設備	421
5. 関係ある帳票様式	463
6. 初級情報処理技術者育成・時間配分表	485
索 引	493

第 1 編 總 論

(1) 総 論

目 次

1. 情報処理技術者の育成	1
2. 初級情報処理技術者育成指針の位置づけ	4
3. カリキュラムの編成	11
4. 学習指導計画と指導法	17



1. 情報処理技術者の育成

コンピュータの進歩とその利用の広がりは、情報の管理と活用に関する新しい技法と適応分野を生み出した。そして、各種産業、行政機関、教育研究機関など社会のさまざまな面で、作業の効率と質を高めている。

他方では、組織の再編成、人的能力と知的資源の開発、情報処理技術者の養成・確保など、従来見られなかった新しい課題をかかえることになり、その解釈の仕方が懸案となって来た。

ここでは、このような推移の中で、情報処理技術者育成という課題がどのような意味を持っているかを述べる。

1.1 情報処理技術者の不足

一般に新しい分野の成長には、研究者、教師、技術者、さらに実務担当者や操作員などに優れた人材が必要である。しかし、優秀な教育の場がなければ、これらの人材をひきつけ、育てることはできないであろう。特に短期間にめざましく発展したコンピュータ・サイエンス、あるいは情報工学 (Informatics) と呼ばれる分野では、人材確保の要求を充足することは容易でない。すでに教育の不均衡によって、コンピュータの成長は脅かされており、教育機会が少ないために、コンピュータに関心をもっている有能な志願者をコンピュータの分野に参加させることができないという状態が生れている。

I F I P副会長 Dr. H. Zemanek は、今後のコンピュータ需要と要員について次のように予想している。

① コンピュータ1台について平均的に必要な要員は、

高級専門家	1人
大学卒業者	3人
高校卒業者	5人
職業コース修了者	7人

の割合である。

② コンピュータ台数は、毎年30%増加し、3年ごとに倍になる。10年ごとに10倍になる。

③ そして近い将来、多くの国でコンピュータ保有台数は、1970年におけるアメリカのコン

ピュータ密度、すなわち2億の人口に対して60,000台の割合に達するであろう。

これに具体的な数字を当てはめてみると、次のようになる。人口百万人の国における到達点は、

コンピュータ	300台
高級専門家	300人
大学卒業者	900人
高校卒業者	1,500人
その他	2,100人

である。1億人の人口をもつ国では、

コンピュータ	30,000台
大学卒業者	120,000人
その他	360,000人

である。

わが国で必要になるのは、30,000台のコンピュータと480,000人のコンピュータ人口である。

この数字を実現可能な値に減少させてみたとしても、なお情報処理技術者を養成するための教育について、緊急かつ巨大な仕事があることは明らかである。

1.2 情報処理教育の質

コンピュータの利用は、1960年代の初期の段階では、日常的かつ断片的な事務処理や計算処理の分野に限られていた。しかし、最近では各種の情報やデータを有機的に利用しながら、"総合的な判断を要する計画・管理・意思決定"に役立てるという高度化の方向を強めている。そしてコンピュータを中心とする情報処理システムは、企業や組織体の中の管理中枢、あるいは頭脳センター的な役割を担いつつある。このような段階では、コンピュータの要員は、もはや単なる技能者としての技術的な能力を持っているだけではつとまらない。いわゆる情報処理技術の習得に加えて、社会科学や社会工学的な物の見かたや、さまざまな業務の実務経験が必要になったのである。こうして、コンピュータ要員は、コンピュータ部門の専門技術者という従来の狭い観念から脱却して、新しい知的技術者への転化を迫られているといえる。

このように考えると、現代あるいは次代の情報処理を支える情報処理技術者は、その教育の初めから、専門技術についての知識とともに、広く関連諸科学に関する基礎知識と、豊かな教養をあわせて身につけて行くことが期待される。加えて絶えず自己学習によって、日々新たに開発

されて行く新しい技術を独力で習得できるよう、研究的な態度や意欲を身につけなければならないのである。

こうした情報処理技術者の量的、質的变化を考えると、情報処理技術についての教育は、旧来のさまざまな技術教育とは異質なものでなければならないことが明らかである。

次のような点が重要である。

(1) 情報処理技術は、かつて工業化を推進するさいに、人材の基底をかたちづけてきた技能とは、異質のものである。従来は、経験から学んだ知識の累積や、徒弟的訓練を経て習得したもので、知識や技術は断片的であり、また一度身につければ長期間有効であった。

新しい知的技術としての情報処理技術は、理論的知識を体系的・組織的に学習しなければ習得できない。そしてたとえ習得できたとしても、そのままでは不断の技術進歩の過程の中で陳腐化してしまうから自己学習によって変化に適応できる能力と、自己革新に対する意欲的態度の育成が同時に要求されるのである。

(2) 情報処理技術者の教育は、それ自身が新しい教育のモデルにならなければならない。すなわち、組織的・体系的に解析できる思考能力や、主体的な学習態度、知的態度の育成を主眼とした基礎教育を中心にして、長い人生のあらゆる時期にわたって絶えず、より高度の、より新しい知識技術の学習ができるという生涯教育の理念の展開である。情報処理についての教育は、その本質からして、そのような新しい教育の先駆的形態の先取りという性格を持たなければならないのである。

(3) 情報処理技術についての教育を効果的に行なおうとすれば、高校、高専、大学などのいわゆる学校教育と、企業内教育、各種講習会、各種学校、さらには放送教育など他のさまざまな教育システムとの間に、有効適切な機能分担や協力関係ができていなければならない。しかし生涯教育という観点から考えると、その中でも、企業内教育は特に重要であり、企業内教育の発展と充実が望まれるのである。

2. 初級情報処理技術者育成指針の位置づけ

文部省では、昭和46年度から高校の工業教育、商業教育において、一般学科の中で情報処理教育を充実させるほか、情報処理技術についての事項を専門に教える情報処理学科を新設して、技術者の養成をはかろうとしている。

工業教育における情報処理学科（情報技術科）の目標は、「電子計算機に関する知識と技術を習得させ、電子計算機を利用する工業生産の各種業務および電子計算機を生産する各種業務に従事する者を養成する」となっている。

商業教育を受ける生徒に対しても、将来産業経済を担う一員として、産業経済界に普及しつつあるコンピュータについての知識を持たせることが必要である。そこで商業教育における情報処理学科として情報処理科が設けられる。情報処理科では、「事務および電子計算機利用に関する知識と技術を習得させ、情報処理に関する業務に従事する者を養成する」ことになっている。普通高校における情報処理教育も拡充されつつある。

たとえば、昭和48年度から、普通課程の数学科に、「電子計算機と流れ図」などの領域を設けることが予定されている。

初級情報処理技術者育成指針（以下本育成指針という）は、その内容はほぼこれら高校における情報処理学科の教育と並ぶものであり、企業内において情報処理についての職業教育を行なうときの指針として作られたものである。

本章では、本育成指針の対象、教育の目標・水準、情報処理教育を担当する教師に必要な資質などについて述べる。

2.1 初級情報処理技術者育成指針の対象者

本育成指針の対象者は次のようなものである。

- (1) 高校普通科2年修了程度の学力を持っている。
- (2) 情報処理技術についての知識が全くない。
- (3) コンピュータを道具として利用することを考え、主担当者の補助者として情報処理業務を遂行することが予定されている。

情報処理の分野は、非常に応用的な分野で、それにたずさわるものには幅の広い知識、分析力や判断力、応用力などが要求される。特定の分野に詳しい知識を持っていたとしても、それが幅広い知

識と調和していなければ有力な武器にはならない。本育成指針で前提としている高校普通科2年修了の学力とは、次にあげるような基礎的な学力が高校2年修了の学力に達しているということである。

国語……読む力、書く力、発表する力、鑑賞評価する力

数学……考える力、計算する力、

理科……現象を観察する力、生活に活用する力、数学を応用する力

英語……原書を読む力、書く力

社会……発表する力、その他

本育成指針では、このような基礎的な学力があれば特に適性は問わない。情報処理技術者としての適性があるかどうかを見るために、適性検査が広く行なわれているが、初級情報処理技術者を育成するために、適性検査が重要であるとは思われない。本育成指針では、基礎的な学力だけを問題にしているのであって、情報処理についての知識は求めている。情報処理技術については未知であることを前提にしてカリキュラムの編成を考えている。

本育成指針では、初級情報処理技術者の育成を目標にしているが、初級情報処理技術者とは何であろうか。中級情報処理技術者、上級情報処理技術者との関連で考えてみよう。

コンピュータを使って事務処理を機械化することを例にとってみよう。そこには、

- ① 関連部門と折衝しながら、機械化の範囲や時期、方法など基本的なことからについて方針をまとめる作業。
- ② その方針に従って機械化の構想を具体化して設計する作業。
- ③ プログラムを作る作業。
- ④ 機械化が実行に移されたあとのコンピュータ操作など日常的な作業。

などがある。

初級情報処理技術者は、③については、適当な技術的指導を受ければ、独力で作業を担当することができるであろう。②については、主担当者のもとで、指導助言を受けながら補助者として、コンピュータを活用した情報処理システムを設計し、具体化する作業にあたることができる。

初級情報処理技術者は、将来実務経験を積み、中級、上級の教育を受講することによって、中級情報処理技術者、あるいは上級情報処理技術者として成長して行くことが期待される。

上級情報処理教育のための上級情報処理技術者育成指針は、「上級情報処理技術研修ガイドブック」として(財)日本情報処理開発センターからすでに刊行されている。中級情報処理教育のための中級情報処理技術者育成指針も、近い将来に作成されることが望まれている。

2.2 情報処理技術者に期待される資質

一般に技術者に必要なこととして、

- ① 何を作るか、何を期待されているかなど、目的や対象を考える。
- ② 方法や手段を工夫する。
- ③ 試作や実験を行なう。
- ④ 成果を得るまで努力する。
- ⑤ 結果を評価・反省し、改善のためのデータをとる。

などをあげることができる。

このような点を考えると、情報処理技術者には次のような資質が必要であろう。

(1) 基本的能力

- (a) 体力と精神力：体力と精神力は技術者に一般的に要求される資質である。
- (b) 思考力：情報処理の分野では、とりわけ「考える」ことが強く要求される。したがって教育では、思考学習を重視しなければならない。この場合、単純な条件のもとでの思考から、むずかしい課題へと、順次高度な思考学習へ移るよう指導して行くことが必要である。
- (c) 創造力と応用力：情報処理技術者は、常に新しいものを創造しなければならないから、創造力と応用力は、技術を支える根源である。
- (d) 記憶力と理解力：進歩の早い情報処理技術の分野では、新しい技術を体得し、活用し、応用するために、豊かな記憶力と理解力が必要である。記憶力は思考・創造などの基盤であるから、理解して正しく記憶するという、学習の態度が望まれる。
- (e) 批判力・判断力、反省力：情報処理については、いろいろな考え方や処理の仕方が存在しており、最善の方式を見つけることがむずかしい。したがって、とりわけ批判したり、判断したり、反省したりしながら向上をはかることが必要である。
- (f) 指導力：情報処理の分野での作業は、考えたことを発表し、承認をうけて、実行するという過程の繰返しであるといってもよい。したがって、重要なのは、積極性であり、指導力である。この資質は特に中級、上級の技術者に対して強く要求される。

(2) 態度と習慣

- (a) 意欲：情報処理は新しい分野であり、将来の産業経済社会を支える重要な分野であるから、情報処理技術者をめざす人たちが、学習や勤労に対して誇りや意欲を持つことは容易であろう。新しい分野を担当する人たちに求められるのは、はつらつとした態度、積極的な姿勢、知識を吸収し、仕事を仕上げるためのあくなき意欲である。

- (b) 忍耐力と持久力：高度の忍耐力と持久力の必要性は、コンピュータと接したことのある人ならば、だれでも痛感するところである。それは可能性を現実のものとするために、また技術の進歩のために不可欠である。
- (c) 向上心と謙虚な態度：情報処理の分野では、研究開発の成果を利用しあうことが多い。また個人の力がいかに微弱であるかを知ることであろう。自己のいたらなさ、誤りをいち早く自覚し、先人の業績に敬意を払い、向上心をもって、あとに続く人に対して進んで貢献する態度が必要である。
- (d) 自主性と協調性：情報処理の分野では、グループで作業することや、関連部門との人的な接触が多いので、協調性が重要である。また、果さなければならない使命を認識し、進んで物事を解決し、自主的に責任を持って作業を進める態度と習慣が望まれる。
- (e) 社交性と独立性：多くの人と積極的に交わり、知識を広くすると同時に、他に依存しないで、習得した知識を応用し、技術を向上させるゆとりがほしい。

2.3 初級情報処理技術者育成指針の目標、水準

本育成指針が目標にしていることの1つに、前節で示したような資質の向上があるが、ここでは教育目標をもっと具体的に述べておこう。

2.3.1 具体的な教育目標

本育成指針の目標は、情報処理技術者に対して期待される初級の技術を習得させることにある。ここで情報処理技術者とは、コンピュータを用いて情報の整理、加工、蓄積、検索などを行なうことを目的として、システムの分析や設計、プログラムの設計や作成などを行なうものという程度の意味で考えている。

情報処理についての初級の技術といえば、コンピュータの知識が中心になるであろう。コンピュータを道具として使いこなせる力を持たなければ、情報処理を考えることはできないからである。そこで本育成指針では、コンピュータを中心にしながら、コンピュータを理解するために必要な、データ処理についての最低限の周辺知識を教えることになる。本育成指針では、次のようなことがらについての教育を期待している。

(1) ソフトウェアの基礎知識

プログラムの基本概念、流れ図、汎用プログラム、ファイルの概念などについて教える。

(2) プログラムの作成能力

プログラム設計書を見ながらプログラムを作れる能力を身につけさせる。

(a) JIS水準3000FORTRAN

(b) 分類、報告書機能、大記憶の処理などの機能を除いたCOBOL65

のうち、いずれかの言語を使って、プログラム設計書を見ながら、一通りのプログラムが作れるようになることを期待する。

(3) ハードウェアの基礎知識

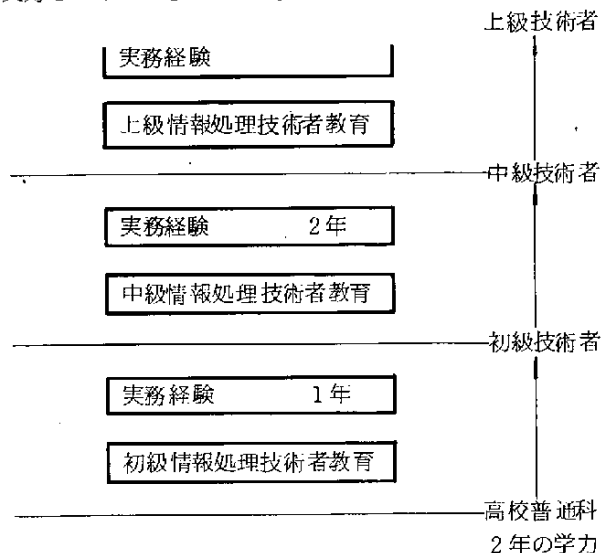
入出力装置、制御装置、演算装置、記憶装置について、種類や機能、特徴などを教える。

(4) 関連知識

情報処理の基本概念や、コンピュータの適用分野についての初歩的な知識を教える。特定の業務に密接に関連する知識は教えない。たとえば技術計算を主とする業務に必要な数値計算の知識には、本育成指針では触れない。必要であれば別に教育する必要がある。統計学の応用なども、本育成指針による教育修了後、別に習得することを期待している。

2.3.2 中級、上級情報処理技術者育成指針との関係

初級の教育を修了したあと、さらに次のような教育を受けて、実務経験を積み、専門的な技術者としての実力をつけることができる。



2.3.3 情報処理技術者試験との関係

昭和44年度から情報処理技術者試験が行なわれているので、技術者試験との関係について触れておこう。

本育成指針にもとづく教育を修了して実務につき、プログラム作成の経験を1年程度積み、

情報処理技術者試験第2種相当の水準に達することを目標としている。

2.3.4 教師となる人に望ましい資質

教育活動では、教師の良し悪しが教育効果に大きな影響を与える。

そこで、ここでは教師に望まれる資質について、

- ・ 一般的資質
- ・ 技術力
- ・ 教育力

という3つの面から考えてみることにしよう。

(1) 一般的資質

教育職員免許法では、教師の一般的資質について常識人としての最低限のわくを定めているが（注1）、望ましい資質という点から見ると、次のような点が重要である。

(a) 教育に対する情熱

教育活動の成果は、教師個人の努力に依存するところが大きいので、教師が教育に情熱を持っているということが大切である。

(b) 尊敬される人格

教師が生徒に信頼され、尊敬されることは教育効果の上で重要である。信頼されたり尊敬されるためには、技術力を持つだけでなく、人格的にも優れていることが必要である。

(c) 積極性

教育活動には、本来、日々創意や工夫を加え、改善して行くという積極的な姿勢が必要で、情報処理についての教育では、特に積極性を持つことが重要になる。教育の内容が新しい分野の技術であり、速い速度で変化している。したがって、常に最新の技術の動向を知り、それを完全

（注1） 教育職員免許法では、次の項目に該当する者には、免許を授与できないとしている。

- (a) 18歳未満の者
- (b) 高校卒業と同等以上の資格をもたない者
- (c) 禁治産者や準禁治産者
- (d) 禁錮以上の刑に処せられた者
- (e) 日本国憲法施行の日以降、政府を暴力で破壊する主張を持つ政党や団体を結成したり、加入した者

に自分のものとして教育に取入れるという努力が求められるのである。

(2) 技術力

技術力については、情報処理技術者試験第1種の合格者(注2)とほぼ同等の実力を持ち、かつ実際に実務経験を持っていることが望ましい。

(3) 教育力

教育職員免許法施行規則では、教師に教育力をつけさせるために、次のような教育に関する専門科目の修習を義務づけている。

- (a) 教育原理
- (b) 教育心理学、青年心理学
- (c) 教科教育法
- (d) 教育実習

したがって、本育成指針にもとづく教育を担当する教師に対しても、それと同等の配慮をし、教育を担当させるのに先だって、教師としての訓練をすることが望ましい。

(注2) 第1種情報処理技術者に対しては、次のような分野の知識と技術力が要求されている。

- (a) ソフトウェアの知識
 - (Ⅰ) コンピュータによる問題処理手順、流れ図、プログラム記憶方式
 - (Ⅱ) ファイル
 - (Ⅲ) プログラミング言語と汎用プログラム・パッケージ
 - (Ⅳ) オペレーティング・システムの基本概念
 - (Ⅴ) オンライン方式についての基本概念
 - (Ⅵ) 文書化
- (b) プログラム設計について
 - (Ⅰ) 入出力データ、ファイル、コードなどの設計
 - (Ⅱ) プログラム構造の設計
 - (Ⅲ) デバッグ方式の設計
 - (Ⅳ) チェック方式の設計
- (c) プログラミングについて
プログラム作成の知識と経験
- (d) ハードウェアの知識について
 - (Ⅰ) 入出力装置、制御装置、演算装置、記憶装置など
 - (Ⅱ) データ伝送、端末機器の種類・機能・特徴など
 - (Ⅲ) システムの構成と効率
- (e) 情報処理についての一般的知識

3. カリキュラムの編成

カリキュラムは、教育目標を達成するための教育計画である。カリキュラムを編成するときには、次のような手順が必要になる。

- ① 教育の具体的な到達目標を決定する。
- ② それぞれの教育目標を達成するのに必要な指導内容を明らかにする。
- ③ 指導内容をさらに吟味して、カリキュラムを構成する各項目を選定する。
- ④ 最後に、各項目の適切な配列と、時間数を決める。

以上によって特定の環境や条件下で実施できるカリキュラムが編成される。

ここでは、初級情報処理教育における指導内容、教育の性格、項目の選定の仕方、カリキュラム編成例などを述べる。

3.1 指導内容

一般に教育内容を選定するにあたっては、受講者が将来従事する職種や作業内容について詳細な資料を得ておくことが必要である。そして教育に対する期待や要望を十分理解した上で、指導内容を抽出することである。また指導内容を定めたあとも、情報処理部門の仕事の内容が絶えず変化し、教育方法や教育技術も時代の推移によって変るものであることに留意し、学会誌、コンピュータ業界の機関誌、研究所の機関誌、その他の専門誌、企業内の情報処理技術者教育の実例などを参考にしながら、常に改善を加えて行くことが望ましい。

この場合、教育に対する期待を調査し、それをできるだけカリキュラムに盛り込もうとすると教えなければならない知識や技能が非常にたくさんあるように思えて、その取捨選択に困る場合が出てこよう。そこで、第1に「基礎的な知識技能とはどんなものであるか」をよく吟味して、正しい理解を持つことであり、第2に教育の具体的目標を十分検討して、「指導の重点をどこに置くか」を見きわめることである。

ここでは、本育成指針をもとづく教育の標準的な項目と指導時間を示す。その内容はそれぞれの教育の場の状況に応じて吟味し修正できる。

標準的なカリキュラムの内容

項	目	標準時間数
基礎編		200時間
	データ処理の基礎	50 "
	事務とデータ処理	15 "
	機械によるデータ処理	16 "
	データ収集	19 "
	コンピュータのハードウェア	50 "
	ハードウェアのあらまし	10 "
	入出力装置	14 "
	記憶装置	13 "
	演算・制御装置	13 "
	コンピュータのソフトウェア	50 "
	プログラムのあらまし	17 "
	プログラミング・システム	15 "
	言語のあらまし	18 "
	コンピュータの利用	50 "
	システム設計	40 "
	コンピュータの応用	10 "
実習編		300時間
	プログラミング	220 "
	FORTRAN } どちらかを	
	COBOL } 選択	
	データ準備	30 "
	コンピュータの運用	50 "

基礎編では、情報処理の基礎的な知識を体系的に教えることを期待しており、200時間を予定する。実習編では主としてプログラミングの技術を修得させる。技術的な分野で実務につく場合と、事務処理の分野で情報処理業務にたずさわる場合とでは、使用するプログラミング言語やプログラミング技法が異なるので、FORTRANとCOBOLは、必要に応じてどちらか一方を

選択させて、教えることにする。実習編では、できるだけプログラミングの実習に時間をかけることが望ましい。

3.2 教育の性格

すでに述べたように、このカリキュラムでは、職業教育の場での初級情報処理教育について標準的な項目を示し、それぞれについて時間数を示してある。しかし、本質的には教育内容を統一し画一化することはできないはずである。職場の状況がさまざまであるし、受講者の技術的な水準も、時間や場所によって異なるからである。講習にさく時間についても、ある企業では一定期間全勤務時間をさいて、集中的に教育を行なうことができるかも知れないし、他の企業では、午後の時間だけとか、勤務時間外を使ってかなり長期的なコースを設けなければならないかも知れない。時間的な制約で場合によっては、短縮コースを行なわざるを得ないかも知れない。

このカリキュラムは、そうした具体的な事例を想定して作られてはいない。初級情報処理教育はどうあるべきかという観点から、標準的な内容を示しているにすぎないのである。したがって教育を企画するものは、それぞれの場における具体的な教育目標に照らして、重点の置きどころを定めたり、教育目標を達成するために工夫をこらすなど個々の努力が期待される。

また教育の目標が企業内で実務にたずさわっているものを直接の対象にしているので、教育が知識偏重にならないよう、実地に役立つ技能面の指導に重きをおいている。教育の中での例題の取り上げ方や、演習の仕方などは、かなり教師の裁量にゆだねられているが、教育の性格から考えて、できるだけ実務の中で実際に採用されている原理・原則や処理方式を例示して行くことが望ましい。

具体的な事例に触れながら、情報処理の概念や役割について基本的なことがらを正しく理解することができれば最も望ましいといえるだろう。

3.3 コースの選定

現在では、情報処理業務にたずさわっていたり、情報処理に関心を持っているものには、必ずしも十分とはいえないが、さまざまな教育の機会がある。主なものをあげれば次のとおりである。

- (1) 高校の工業教育の分野における機械・電気・電子・化学・建築・土木・自動車・造船・紡績・色染などの学科において拡充しつつある情報処理の基本技術に関する教育
- (2) 高校の工業教育の分野に昭和38年から生れた電子科におけるコンピュータに関する基礎的な教育

- (3) 高校の工業教育の分野に新設された情報技術科における情報処理技術に関する教育
- (4) 高校の商業教育における商業科、事務科、経理科などに新設されつつあるコンピュータを用いての情報処理に関する教育
- (5) 高校の商業教育の分野に新設されつつある情報処理科における情報処理教育
- (6) 電子計算機専門学校などの各種学校
- (7) コンピュータ・メーカーが提供しているユーザに対するコンピュータ教育
- (8) 特定のソフトウェア会社や民間の教育機関が行なっているコンピュータ教育
- (9) テレビやラジオのコンピュータ講座
- (10) コンピュータに関する講習会、講演会、ゼミナールなど。
- (11) コンピュータに関する入門書や独習書

教育の受講者は、これらの教育の機会に多かれ少なかれ触れているであろう。したがって、本育成指針に盛られている内容について、すでに断片的な知識や、場合によっては、部分的に相当な知識を持っているかも知れない。しかし、それらの知識を前提にして、安易に、カリキュラムの内容を削除すべきではない。受講者の持っている知識が完全であるという確信なしに特定の項目を削除することは、教育効果を半減することになるであろう。本育成指針が目標にしているのは、コンピュータや情報処理についての断片的な知識ではない。本育成指針の目標は、情報処理についての基本的な概念と技法を身につけ、それを実際の場に応用できる力、さらに中級、上級と技術水準を高めて行く力を身につけることである。

したがって教育の場で、項目を決め、時間数を割当てるときには、指導内容の適正な配列と、各項目の指導内容の相互の関連、各指導内容の学習時間の調整などに、十分な配慮が欲しい。また実習施設や設備、教員組織なども考えて、いたずらに理想に走り、実施上困難を生じることのないよう留意し、しかも効果的な教育の仕方を考えなければならない。

また、すでに述べたように、カリキュラムは、1度これを編成すれば、それでよいというものではなく、科学技術の進歩、産業界の実態、施設・設備、教員組織の変化に応じて検討を加え、教育効果をあげるよう不断の努力が必要である。

3.4 カリキュラムの編成例

次にカリキュラムの編成例を示す。

(1) 標準コース

期 間 4カ月 17週間

時間割 1日6時間(午前2時間30分、午後3時間30分)

1週30時間(月曜日～金曜日)

土曜日は、自習にあてる。

編成例

週	午 前			午 後		
	日数	時間数	内 容	日数	時間数	内 容
1～4	20	50	基礎 データ処理の基礎	80	280	実習内容は、生徒の理解度、実習の環境条件などによって決める。 1例をあげると次のとおり。 1.データ準備 2.プログラミング (基礎的なもの) 3.コンピュータの運用 4.プログラミング (総合演習的なもの)
5～8	20	50	基礎 コンピュータのハードウェア			
9～12	20	50	基礎 コンピュータのソフトウェア			
13～16	20	50	基礎 コンピュータの応用			
17	5	12.5	総合演習、まとめ	5	17.5	総合演習、まとめ

合計 17週、85日、510時間

(2) 短縮コース

受講者の基礎知識、実務の要請、教育環境や教育施設・設備の条件などによって、標準コースを分割して行なったり、場合によっては内容の一部を省略して、教育期間を短縮することなどが可能である。

ただし、教育期間を短縮する場合には、それによって教育効果が低下しないかどうかを十分検討する必要がある。

前テスト(pre-test)を行なって生徒の知識を確かめたり、前もって読んでおくべき本や準備しておくべきこと(prerequisite)を指定して、受講者の水準をそろえておくことなどの配慮が必要である。

(3) 実習計画

実習編のカリキュラム編成をどうするかは、本育成指針にもとづいてカリキュラムを編成する場合、特に重要である。

実習編では、講義と演習が交互に行なわれるので、それをどう組合わせるか、演習をいかに効果的に行なうかが重要である。演習では、いろいろな施設や設備を使用する。それらの施設・設備をむだなく有効に使用できるような演習の進め方が必要である。プログラミングの演習で使用する施設は次のとおりである。

使用する施設 プログラミング実習室

情報処理実習室

せん孔実習室

プログラミング実習室は、講義、プログラミング、デバッグなどの演習のために使われ、使用時間が最も長い。

情報処理実習室の中心は電子計算機室である。電子計算機室では、コンピュータをいかに能率的に使うかが重要である。

せん孔実習室は、紙カードをせん孔するために使う。けん盤せん孔機(key punch)の台数が十分でないときは、せん孔実習室が一番待ちのできる可能性が大きい。

生徒数を30名と考えて、1グループ3名で演習のグループを編成したとすると10グループができる。その場合10グループがどのような作業をしたら3つの実習室を最も有効に活用できるであろうか。重要なことは、コンピュータの使用時間をどうスケジュールするかである。

実習の流れは、

プログラミング→紙カードにせん孔→コンピュータ処理→出力データの解析、検討→報告書の作成

となる。この流れが何度も繰返されて演習は進行するが、それぞれの段階の作業が全体の中でどの程度の割合を占め、どの程度の時間を要するのかを検討して、演習時間や、コンピュータ時間の割当てなどを決める必要がある。早く演習が完了したグループには、他のグループの指導に当らせたり、あるいはより正確な、読みやすい報告書を完成させるよう、報告書作成に時間をかけさせるなどの配慮が必要になる。

4. 学習指導計画と指導法

学科の具体的目標が設定され、その目標を達成するためのカリキュラムが編成されたら、次にそのカリキュラムを実施するための学習指導計画を作らなければならない。そこで、ここでは学習指導計画を立案する方法と、その場合に考慮しなければならない学習指導法について述べる。

4.1 学習指導計画

定められたカリキュラムに基づいて教育を行なう場合、各科目の学習内容の展開に適切さを欠くことがあれば十分な学習指導はできない。そこで、教育を始めるに先立って学習指導計画を作成しておくことが重要になる。

学習指導計画には次の3つの段階がある。

学習指導の概要計画

学習指導の詳細計画

学習指導案

次にそれぞれについて簡単に説明する。

(1) 学習指導の概要計画

概要計画を立案するにあたっては、この教育が必ずしも学校教育のように、教育環境や教育条件が十分でない企業内を主体にして行なわれることを考えなければならない。

次の点を十分に考慮すべきである。

- (a) 企業内では、教育に要する時間は非生産的な時間であり、できるだけ短時間に切りつめたいという考えから、教育は実務によって圧迫されやすい。
- (b) 教育の場所や教育環境にあわせた計画が必要である。
- (c) 実務にたずさわりながら、教育を受ける場合には仕事との間のけじめをつけるための配慮が必要である。それには次のようなことが考えられる。
 - (i) 4カ月間集中して全コースを行なう。
 - (ii) 集中教育(主として基礎編)、実務、集中教育(主として実習編)の組合せで教育を行なう。
 - (iii) 午前の実務を行ない、午後集中教育を行なう。

学習指導の概要計画を立案するにあたっては、基礎編、実習編を次の単位に分け、各単位の具体的目標、学習指導の内容、指導の手順、重点の置き方、配当時間、他との関連などを検討し、実施時期を決める。

基 礎 編	データ処理の基礎
	コンピュータのハードウェア
	コンピュータのソフトウェア
	コンピュータの利用
実 習 編	プログラミング (FORTRAN)
	プログラミング (COBOL)
	データ準備
	コンピュータの運用

指導の順序を決めるときには

- (a) 基礎編と実習編は並行して進めることができる。
- (b) 実習編のプログラミングとデータ準備、コンピュータの運用は、生徒の理解度、実習の環境条件などを見ながら、実施の順序を決める。また、実習の進行に合わせて並行的に行なうこともできる。

ことを考えておくとよい。

(2) 学習指導の詳細計画

学習指導の概要計画で決めた各単位について、指導時間・指導目標・指導事項・指導内容・指導方法などを記載する。

次のような表を作るとよい。

学習指導の詳細計画の1例

項目名	実習プログラミング FORTRAN			
中項目	データの型	第 9 ～ 16.5 時間目	時間数 7.5	時間数 7.5
目 標	1. FORTRAN で使用するデータには、実数型と整数型があることを理解させる。 2. 定数と変数の概念をはっきりつかませる。 3. 配列に対する理解を深める。			

小項目	主な指導事項	指 導 内 容	指 導 方 法	指導時間
1.定数	(1)整定数	整定数の表わし方	黒板などに書いて説明する。	0.5
	(2)実定数	(a)固定小数点数の表わし方 (b)浮動小数点数の表わし方	黒板などに書いて説明し、 ある数を固定小数点や浮動 小数点で書かせてみる。	1
2.変数	(1)整変数	(a)整変数とは何か (b)整変数の表わし方	記憶場所の図を書いて説明 する。	1
	(2)実変数	(a)実変数とは何か (b)実変数の表わし方	記憶場所の図を書いて説明 する。	1
3.配列	(1)配列の概念	(a)配列の定義 (b)配列, 配列要素, 添字 などの名称について (c)配列を利用する利点	ベクトルや行列, びんごの 目の例などで図を用いて説 明する。	1
	(2)配列の種類	(a)定数型 (b)実数型	黒板に書いて説明する。	1.5
	(3)添字	(a)整定数の添字 (b)整実数の添字	黒板に書いて説明する。	1.5

(3) 学習指導案

学習指導の詳細ができたなら、教育する小項目ごとに指導しなければならない指導内容と指導手順を決める。それが学習指導案である。

学習指導案では、目標、内容を明らかにし、準備すべきこと、導入、展開、整理の内容、評価の着眼点などを記載する。

次の図はその例である。

学習指導案の1例

指導小 項 目	定数(2)実定数	時間 6 0 分	指導中項目	データの型
内 容	1.固定小数点数の表わし方 2.浮動小数点数の表わし方			

善するために欠かせない手段である。試験やそれに準ずる手段は、生徒を評価するだけでなく、教師の反省資料にもなるはずである。

教育は、そのようなフィードバックの手段によって改善される。カリキュラムを作成するにあたって、宿題を課するとを前提にするかどうかは、それぞれの職場の状況による。しかし、たとえ宿題を前提にしない場合でも、生徒に対しては、できるだけ復習を行なわせるべきである。積極的な生徒には予習を期待してもよい。情報処理技術については特に知識の積重ねが重要なので、いったん足踏みすると、なかなか追いつけないことをよくわからせるべきである。

生徒が全くの初心者である場合、教育期間を6カ月に延長して考えてもよい。また実習の最後の段階では、宿泊研修を行なって、報告書作成や、発表なども含めて、総合的な訓練を行なってもよい。

第 2 編 各 論

基 礎 編

第 1 部 データ処理の基礎

教育の目標

社会経済の発展とともに、官公庁、民間企業などの組織や相互の関係は複雑になり、活動範囲が拡大するにつれて事務量がふえ、事務処理に要する経費が増大した。また情報を選択、加工、分析して、有効な管理資料や経営情報を得ることが切実な課題になった。

第 1 部ではこれらの背景を考えながら、データ処理あるいは情報処理の意味を説明し、第 2 部以降の導入とする。

最初に事務とはどのような作業をいうのか、情報とは何か、データ処理とは何かなど、事務とデータ処理についての基礎的な概念を説明する。次にデータ処理を機械化する意義、機械化したときのデータ収集方法などを説明し、データ処理の基礎になる知識を習得させる。

内容のあらまし

内 容	時間数
事務とデータ処理	
情報とデータ処理	3
事務の機能	6
データ処理の機能	6
機械によるデータ処理	
データ処理方法改善の必要性	3
データ処理方法の歴史	5
機械による処理	8
データ収集	
データ収集方法のいろいろ	3
紙カード	4
紙テープ	4
その他のデータ収集媒体	5
データの変換	3

第1章 事務とデータ処理

この章では次の用語を教える。

会計機、加算器、紙カード(カード)、紙テープ、検孔機、けん盤せん孔機(キーパンチ)、コーディング、照合機、情報、情報処理、突合せ、データ、データ処理、なまデータ、複写(する)、技術分野、事務分野、マーク・センス・カード、変換、分類、データ通信、パンチ・カード・システム、配線盤、制御

目 標

われわれは日常、「データ」とか「情報」とかいうことばを深い考慮も払わずに用いている。そこで、はじめに「データ」と「情報」ということばの意味を説明し、「データ処理」の概念を明確にする必要がある。

また、データ処理の対象になる「事務」とは何か、「事務」と呼ばれる作業には具体的にどのような作業が含まれるのかを説明する。事務処理では、伝票、帳簿、報告書などが重要な意味を持っている。したがって、伝票の起票、検査、記帳、報告書の作成などが事務の大半を占め、そのために多くの人力が投入されている。これらの作業を人手のかわりに機械で行なう方法は、昔から研究されてきた。こうした事務処理改善の努力を説明する。

内 容

1.1 情報とデータ処理

データ、情報、データ処理、情報処理などの概念を説明する。

情報処理用語のJIS規格では、最初にデータと情報の定義をしている。しかしここではJIS規格のような厳密な定義をするよりも、できるだけ概念をくだいて具体的に説明する方がよい。選挙の投票、在庫量、ガスメータの検針、学校の出席記録、選挙統計、技術性能記録、生産量など、データの具体例をいくつか取上げて説明する。技術分野と事務分野のデータの性質の違いにも触れる。

データ処理と情報処理の例をあげて説明する。

1.2 事務の機能

事務や事務処理について説明する。

次のような内容について説明し、企業活動と事務処理について理解を深める。

- (a) 事務の型
- (b) 事務組織
- (c) 事務処理
- (d) 経理事務

次にそれぞれについて説明すべき内容を述べる。

1.2.1 事務の型

生産部門、販売部門、サービス部門の特徴を説明する。

一般的に各種の業種を上の3つの型のどれかにあてはめることができるし、1業種の中の事務をそれぞれに区分することもできる。

製造業、装置工業、流通部門、官公庁、金融業、新聞・放送・出版、電気・ガスなどの業種の例をあげ、それぞれの事務について簡単に説明する。

企業形態としては、代表的なものとして株式会社を取上げて事務の特徴を説明する。

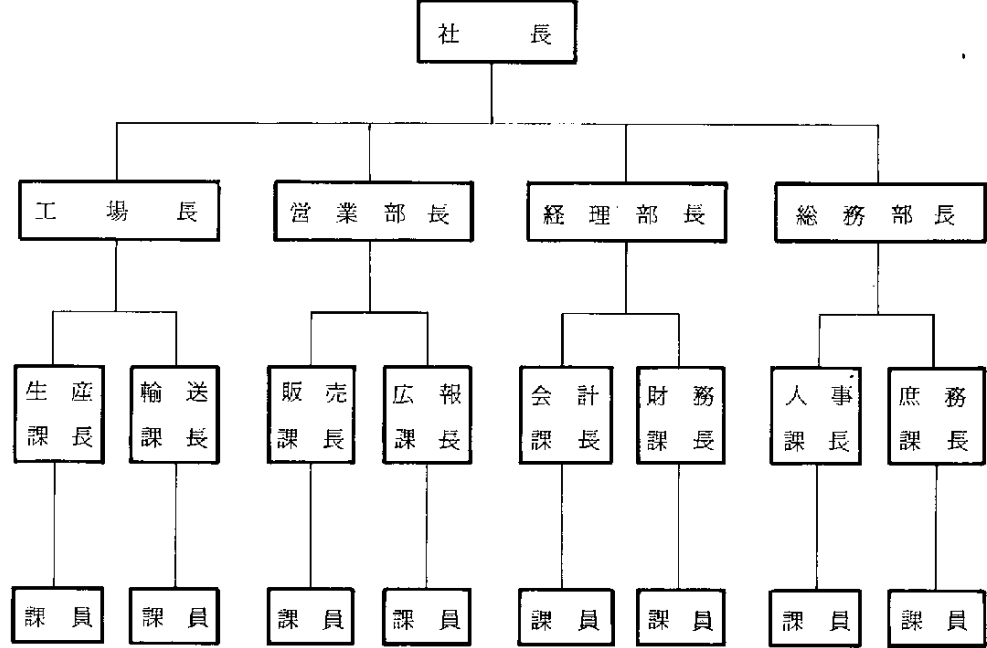
1.2.2 事務組織

次の3つの組織について、特徴を説明する。

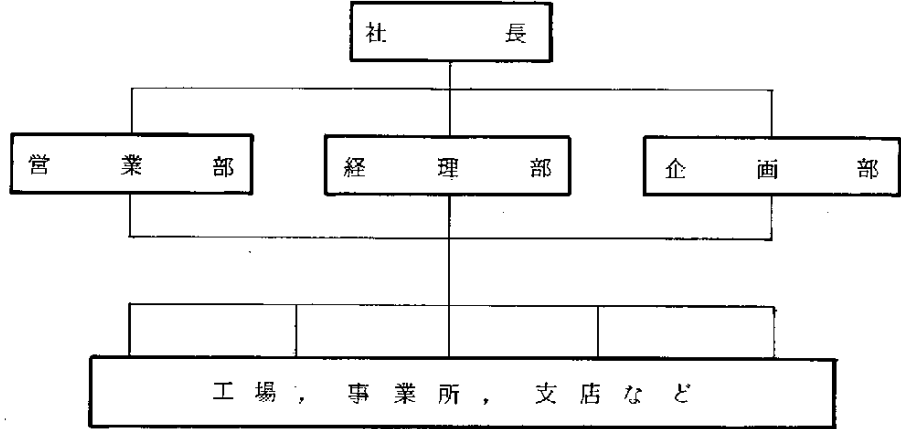
- (a) 直系組織(line organization)
- (b) 職能組織(functional organization)
- (c) ラインスタッフ組織(line-and-staff organization)

次のような図を1例として示すとよい。

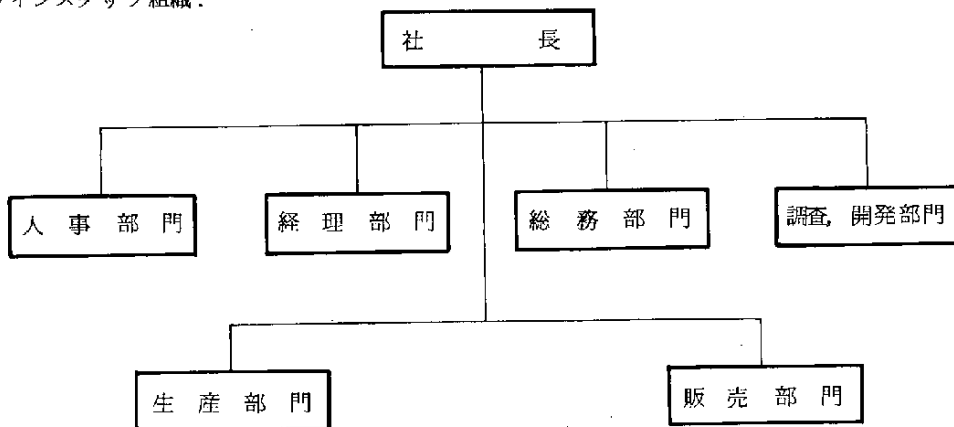
直系組織：



職能組織：



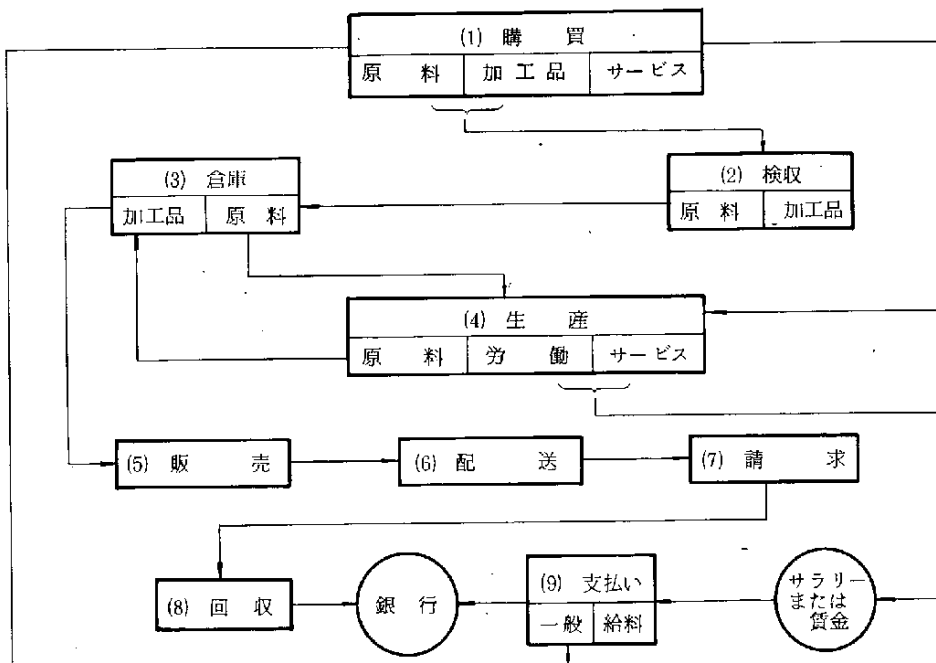
ラインスタッフ組織：



1.2.3 事務処理

生産部門、販売部門、サービス部門はそれぞれ内部にどのような小部門を持ち、事務処理を行っているか、それぞれの小部門での事務処理は、お互いにどのように関連しあっているかを述べる。

特定の部門を例にとって説明する。たとえば、生産部門の例では、次のように9部門に分割して説明するとよい。



この考え方は、商品がどのように現金に変わるかを中心にして事務処理を分割したものであり、販売部門やサービス部門にも当てはまる。

各部門について、取扱う伝票を中心として簡単に説明する。

(1) 購買部門

次の伝票を説明する。

- (a) 購入依頼書
- (b) 見積り
- (c) 注文書

(2) 検収部門

次の伝票を説明する。

- (a) 納品書
- (b) 検査記録

(3) 倉庫部門

次の伝票を説明する。

- (a) 納品書
- (b) 生産指示書
- (c) 原材料出庫依頼書
- (d) 受注伝票
- (e) 在庫記録書
- (f) 購入依頼書

発注点にも触れる。

(4) 生産部門

次の伝票を説明する。

- (a) 生産指示書
- (b) 出庫依頼書
- (c) 製造過程で作成される記録や報告書

(5) 販売部門

次の伝票を説明する。

- (a) 販売伝票
- (b) 販売指示書

(6) 配送部門

出荷指示書の説明をする。

(7) 請求部門

送り状と請求書について説明する。

請求書は、手作業、手作業と機械作業の組合せ、パンチ・カード・システムや、コンピュータなどによって作成されるが、どの方法を使うかは、作業の量で決る。次のような手順をふんで請求書が作成されることを述べる。

(a) 目的別に項目を記入

(b) 単価を記入

(c) 単価と個々の数量を掛ける。

(d) 合計計算

(e) 税金、輸送費、その他の費用

(f) 請求書に記入またはタイプして、必要な枚数のコピーを作成する。

コピーは、客先、販売員や販売部門などに送る。

(8) 回収部門

次の伝票を説明する。

(a) 勘定明細書

(b) 領収書

(9) 支払い部門

人件費と部品費について説明する。賃金台帳、給与支払い領収書、物品費支払い領収書を説明する。

1.2.4 経理事務

前節で説明した事務処理と関連させて、経理事務の概要を説明する。

次のような順を追って説明するとよい。

(1) 起票

(2) 仕訳

(3) 記帳

(4) 試算表の作成

(5) 財務諸表の作成

貸借対照表と損益計算書について説明する。

次のような図を用いるとよい。

貸借対照表

資 産 の 部	負債及び資本の部
流 動 資 産	流 動 負 債
固 定 資 産	固 定 負 債
繰 延 資 産	引 当 金
	資 本 金

損益計算書

売上高
売上原価
一般管理費及び販売費
営業外収益
営業外費用
当期純利益

(6) 決算整理

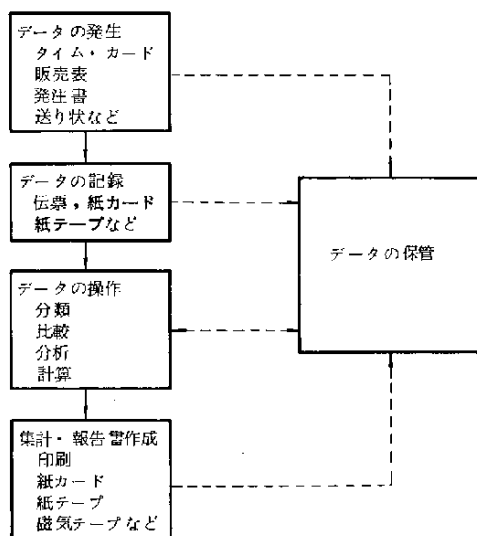
1.3 データ処理の機能

データ処理の過程、データ処理の方法などを実例をあげて説明し、データ処理の機能を理解させる。

データ処理の過程は、手作業で行なう場合もコンピュータを使用して行なう場合も、基本的には同じである。

1.3.1 データ処理の過程

処理の過程を図示すると次のようになる。



(1) データの発生

データ処理の基礎となるなまデータにはどのようなものがあるかを説明し、それぞれのデータの特徴を述べる。

(2) データの記録

一番広く使われているのは、伝票に記録する（記票）方法である。コンピュータでデータ処理をするときは、データは伝票から、紙カードや紙テープにせん孔されて使用される。会計機や加算器による記録、マーク・センス・カード、オンライン・システムの端末機の例などもあげる。

データの記録に関連して次の点を説明する。

(a) 編集(editing)

必要なデータを選択する。

(b) コーディング(coding)

コード付けをする。

(c) グループ分け(classifying)

処理に応じて、いくつかのカテゴリーに分ける。

(d) 変換(conversion)

伝票から紙カード、紙カードから紙テープなどに変換する

(e) 複写(copying and duplicating)

(f) 検証(verifying)

(3) データの操作

次のようなデータの操作について説明する。

(a) 分類(sorting)

手作業で処理する場合は、仕訳とっている。

分類では、ある順番に整理したり、共通の特色を持つものを選び出したりする。

(b) 突合せ(matching)と分析(analyzing)

(c) 計算(calculating)

(4) 集計(summarizing)と報告書作成

毎日、毎週、毎月、毎期などに行なうデータの集計と報告書作成について説明する。

(5) データの収集

データ処理部門について説明する。

コンピュータでデータ処理を行なうためには、まず、データをコンピュータ室に集めなければ

ならない。データを集める方法として、運搬、郵便、テレタイプ、通信回線によるデータ通信などがあることを説明する。集められたデータは変換されて、コンピュータに入力できる媒体に記録される。

1.3.2 データ処理の方法

(1) 手作業による処理

手作業による処理の方法を、例をあげて説明する。生徒に関連の深い事務を取上げて、伝票の起票から、データを集計して報告書を作成するまでの、一連の過程を説明するとよい。

(2) 機械による処理

タイプライタ、加算器、卓上計算機、会計機、^{〔脚注〕} 金銭登録機などの利用について説明する。これらの機械を使用すると、出力として紙テープ、ジャーナル・テープ、紙カードなどが得られ、それを直接コンピュータに入力できることにも触れる。

(3) パンチ・カード・システム (PCS)

パンチ・カード・システムの歴史を簡単に述べ、パンチ・カード・システムによるデータ処理の概要を説明する。パンチ・カード・システムの制御は、配線盤によって行なわれることを述べる。

次の点に触れる。

- (a) けん盤せん孔機 (key punch)
- (b) 検孔機 (verifier)
- (c) 分類機 (sorter)
- (d) インタプリタ (interpreter)
- (e) 照合機 (collator)
- (f) 集団複写せん孔機 (reproducer)
- (g) 計算せん孔機 (calculator)
- (h) 会計機 (accounting machine)

(4) コンピュータによる処理 (electronic data processing)

詳しい説明は第2章にゆずって、ここでは簡単に紹介する。

〔注〕会計機は、一般に比較的単能な事務機械とPCSにおける印刷作表機械の2通りの意味に使用されているが、ここでは前者を指す（付録3 情報処理用語の会計機の項参照）。

指導上の留意点

第1章では、事務とデータ処理にできるだけ親しませることが大切である。事務用語を覚えさせようとする必要はない。

最初に事務に親しませておくことは、事務処理に対するコンピュータの利用という点についての理解を助けるだろう。

説明はできるだけ実例に即して行なう方がよい。伝票類などは、実物を用意する。

データ、情報、データ処理、情報処理などの用語は、いろいろな意味に使われているので、あまり厳密な説明をしようとするとう理解を困難にすることになりかねない。したがってどのような説明をすれば導入として有効であるかを検討し、導入の助けとなるような説明をくふうする必要がある。

第2章 機械によるデータ処理

この章では次の用語を教える。

EDPS, カード・システム, 出力, 処理, 入力, 入力媒体, 出力媒体, コンピュータ, せん孔
(パンチ)

目 標

情報量の増加とともに、手作業による処理の負担が大きくなり、データ処理を機械化することが必要になった。ここでは、データ処理を機械化する必要性を述べ、手作業で処理するのに比較して、機械で処理すると、どのような利益がもたらされるかを理解させる。

内 容

2.1 データ処理方法改善の必要性

時代の進歩とともに、データ処理の方法も進歩した。特に、最近20年間の発展には、めざましいものがある。データ処理の進歩を促した要因には次のものがある。

(1) データ量の増加

特定の事務を例にとって、データの量の変化をグラフなどで示すとよい。

(2) 事務にかかる費用の増加

事務量の増加は、事務員の増加をもたらし、事務処理にかかる人件費やその他経費を増加させた。そこで経費を節約するために、データ処理を機械化する必要性が生じた。

(3) 正確な情報の要求

同じ仕事の繰返しというのは、辛い作業であり、誤りを招きやすい。したがって、正確さを欠くことになりかねない。正確度を高めるために、機械による処理を考えるようになった。

(4) 必要な時期に必要な情報を得る管理資料や経営資料に対する要求が強まり、手作業では困難な情報の加工、分析が必要になった。

2.2 データの処理方法の歴史

政治や産業の発展とともに処理すべきデータ量が増加し、処理方法も複雑さを増した。データ処理の歴史は、人間がこれらのデータの収集、記録、計算のために有効な方法を見つけようと努力して来た歴史でもある。データ処理がどのように発展してきたかを、次の4つの項目に分けて説明する。

- (a) 記録技術
- (b) 計算の道具
- (c) パンチ・カード・システム
- (d) コンピュータ

(1) 記録技術

よく知られていることがらを取上げて、記録技術の発展の歴史を述べる。取上げることがらとしては、たとえば、表意文字、表音文字などの文字の発明、エジプトにおける紙草(papyrus)、古代の紙にかわる記録媒体、中世の紙、印刷技術の発達、帳簿の発生などが考えられる。現代の技術としては、タイプライタその他の記帳用機械の登場に触れる。

(2) 計算の道具

人間の手足や小石などを使って計算していた時代から、コンピュータを使って処理が行なわれるようになるまでの発展の歴史を述べる。そろばんや中世以降に考えられた各種の計算機など、よく知られている例をいくつかあげて、日本および世界の動向を簡単にたどるとよい。

(3) パンチ・カード・システム

チャールズ・バベジ(Charles Babbage)の計算機

ホレリス(Hollerith)の計算機に簡単に触れ、パンチ・カード・システムの発達の歴史を述べる。MARK-IとMARK-IIについて説明する。

(4) コンピュータ

ENIACに始るコンピュータ開発の歴史とその商用化の過程を述べる。

商用化されたコンピュータについて、第1世代のコンピュータ、第2世代のコンピュータ、第3世代のコンピュータの特徴を簡単に説明する。

2.3 機械による処理

コンピュータによるデータ処理を中心に、機械による処理の特徴的なことを簡単に説明する。

コンピュータについては、第2部以降で詳しく説明するので、ここでは手作業によるデータ処

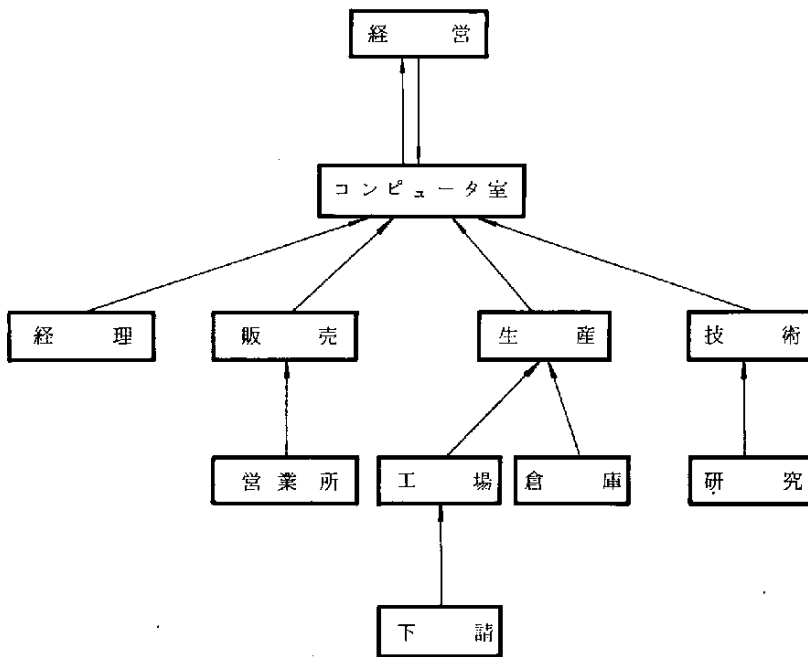
理とどう異なるかを中心に説明するとよい。

コンピュータによるデータ処理の組織、コンピュータによる処理の特徴、コンピュータ処理の実例などを説明する。

2.3.1 コンピュータによるデータ処理の組織

図を示して説明する。

次の図はコンピュータでデータ処理を行なうときの組織の1例である。



2.3.2 コンピュータによる処理の特徴

データ処理過程を、紙カードによる処理(カード・システム)を中心に入力(input)、処理(processing)、出力(output)に分けて説明する。

(1) 入力

伝票に記入されたデータは、コンピュータ室に集められ、けん盤せん孔機によって紙カード(入力媒体)にせん孔される。

(2) 処理

コンピュータにデータを入力して、記憶装置に貯えること、分類・比較・分析・計算・集計な

どのデータ処理を行なうこと、これらの処理はプログラムによって指示すること、などを述べる。

(3) 出力

出力媒体について説明する。

コンピュータによるデータ処理の利点について、次の点を説明する。

- (a) 処理時間が速い。
- (b) 処理は連続的に行なわれる。
- (c) 正確度が高い。
- (d) データ処理を集中し、総合的に行なうことができる。
- (e) 手作業では困難であったデータの加工や分析ができる。

2.3.3 コンピュータ処理の実例

給与計算などの事例を取上げ、手作業による処理と対比させながら、コンピュータ処理の過程を説明する。

事務処理の流れ図を書いて説明するとよい。

流れ図記号の意味を説明する必要はない。

コンピュータによる処理を行なうための準備、手作業からコンピュータ処理への切替え時の処理、コンピュータ処理をしたときの事務処理の変化などにも触れる。

指導上の留意点

この章では、主として事務処理の分野で、コンピュータがなぜ使われるようになったのか、またコンピュータによる処理が手作業と比較して、どう違うのかを理解させることが大切である。

コンピュータということばの同義語として、電子計算機、計算機、EDPSなども広く使われているので、紹介しておいた方がよい。

この章は、第2部以降の導入でもある。導入であるから内容にあまり深く立入るべきではない。詳しい説明はすべて第2部以降で行なう。

データ処理の歴史を説明するときには、写真などをできるだけ多く用意して利用すると有効である。

第3章 データ収集

この章では次の用語を教える。

英数字コード, Xせん孔, カードのけた(欄), カード・コード, 可変長, 固定長, さん孔テープ, 自動データ処理, IDP(集中データ処理), ゾーン, ゾーンせん孔, データ収集, 特殊文字コード, Yせん孔, 磁気テープ, ホレリス・コード, コード, 奇偶検査, 磁気インク文字, 光学文字, ブロック間間隙, カード読取装置(カード・リーダ), 紙テープ読取装置, EDP

目 標

データ処理をコンピュータで行なうためには, コンピュータが扱える形式でデータを集めなければならない。そのようなデータ収集の仕方としていろいろな方法が考えられているが, 代表的なものは, 紙カードと紙テープによるデータ収集である。この章では, この2つの媒体を中心に, 媒体上でのデータの記録の仕方 データ収集の仕方などを説明する。その他のデータ記録媒体の特徴, 用途などを理解させる。

内 容

3.1 データ収集方法のいろいろ

データ収集の自動化を最初に考えたモールスについて簡単に説明する。モールスのコードについて述べる。

データ収集の媒体については, 紙カード以外の発展は遅かった。しかし, IDP(integrated data processing)の概念が導入されてから, 急速に自動化が進んだ。IDPについて, 次のような説明をする。

- (1) IDP(集中データ処理)では, 会計機, 加算器, タイプライタなどで, 記帳, 計算, 集計などをしながら, 同時にその内容をせん孔できる。
- (2) 初期のIDPは, 機械の感知できるコードを媒体にせん孔したが, 最近では, 光学文字認識とか, 磁気インク文字認識なども使われるようになった。
- (3) 紙テープから紙カードなど, 媒体間の変換が可能になった。

(4) IDPの目的について、次の2つのことを述べる。

- (i) データ発生部門で、機械の感知できるコードで、データを記録できる。
- (ii) なまデータ、あるいは自動的に変換されたデータを入力して、すべてのデータ処理を行なうことができる。

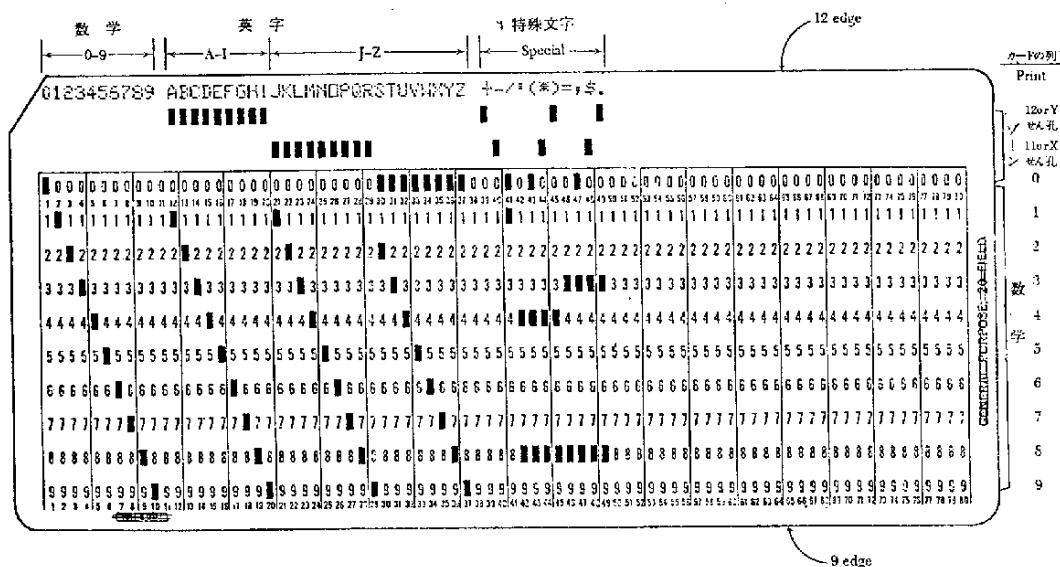
IDPの概念は、自動データ処理(automatic data processing)の考え方に発展し、電子データ処理(EDP electronic data processing)の開発が進んだことを述べる。

データは、ある作業の副産物として作成されるか、あるいは、ある媒体から他の媒体に変換することにより作成される。

そこで、次にデータ収集のために一般的に使われている、コード、媒体、機械などについて説明する。

3.2 紙カード

紙カードは、現在最も一般的に使用されている媒体である。標準的なカード・コード(ホレリス・コード)の説明をする。次のような図を用いて、カードのけた、英数字コード、特殊文字コード、ゾーンせん孔、Xせん孔、Yせん孔などを説明する。



けん盤せん孔機の長所や特徴を説明する。携帯用のカードせん孔装置にも触れる。せん孔された紙カードは、カード読取装置（カード・リーダ）で読込まれる。

3.3 紙テープ

通常の紙カードは80けた（固定長）であるが、紙テープにはけた数の制約がない（可変長）、データを連続して記録できる、などの特徴を説明する。

次のことを教える。

(1) コード

紙テープには、5チャンネル、6チャンネル、7チャンネル、8チャンネルと呼ばれるものがある。ここでは、6チャンネル、8チャンネルを取上げて、それぞれの特徴を説明する。

(a) 6チャンネルのコード

日本で一般に広く使われていることを述べる。シフト・コードについて説明する。

(b) 8チャンネルのコード

奇偶検査（parity check）の機能に触れる。特殊機能文字について簡単に説明する。

(2) 紙テープせん孔機

次のような紙テープせん孔機の特徴や用途を説明する。せん孔された紙テープは、紙テープ読取装置で読取られる。

(a) 自動式タイプライタ（automatic typewriter）

これは、記帳と同時に紙テープにせん孔（さん孔）できる装置である。データがせん孔され記録された紙テープを一般にさん孔テープと呼んでいる。

(b) 紙テープせん孔装置付会計機

紙カードをせん孔するものもある。

(c) 紙テープせん孔装置付加算器

(d) 紙テープせん孔装置付金銭登録機

3.4 その他のデータ収集媒体

次のような媒体を紹介する。

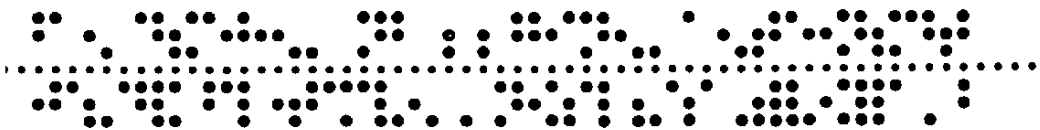
(1) エッジ・パンチ・カード

下のような図を参考にして、次のことを説明する。

(a) コードは紙テープと同じ概念で使用される。

- (b) 紙テープに比べて丈夫であり、紙カードをファイルするのと同じようにファイルできる。
- (c) 繰返し使用される短い情報を記録するのに便利である。

トウキョウト ミナトク シバコウエン 21-1-5
 コホン ショウホウシヨリ カイハツ センター



(2) エッジノッチ・カード (edge-notched card)

最初は、人間の手によるデータ処理のために使用されていたが、現在では機械的に処理できる。
 また、コンピュータの入力のために、紙テープや紙カードに変換できる。

(3) パンチ・タグ (punched tag)

コードは、紙テープの5チャンネル・コードと同じものが使用される。表現できるのは数字だけで、デパートなどで使用されている。タグに書かれているデータを処理するためには、紙カード、磁気テープなどに変換しなければならない。

(4) パーフォレート・クーポン (perforated coupon)

金融業やデパートなどで支払額を記録するために使用されてきた。

(5) バー・コード (bar code)

バー・コードについては、次のような説明をすればよい。これは、プラスチック板に、棒線で数字のデータを表わしたものであり、この棒線の数や打たれている位置によって、固有の数字を表わす。処理するときには、紙カードや紙テープに変換する。

(6) 光学文字 (optical character)

次のことを説明する。

(a) 走査処理を通して光学文字読取装置 (optical character reader, 略してOCR) で読取られる。

(b) 読取りを確実にするために、活字の形は、機械に読みやすいように、独特な形に決められて

(c) OCR活字をタイプライタ、加算器、金銭登録機などに取付けて、OCR印字機として用いることができる。OCR金銭登録機は、ジャーナル・テープを媒体としている。

(7) 光学マーク (optical mark)

用紙に記入されたバー・コードやマークを光学的に読取る。試験の採点などに使われる。

(下圖見本參照)。

注意事項

この答案用紙は完全式に点検で採点しますので、次の事項をよく読んでください。

1. 受験番号は次の所定の欄間をきまたは
H0の留量で書きぬりつゝふすこと。
2. 答案用紙は、折りまけたり、巻いたり、
よごしたりしないこと。
3. 解答は、問題用の留紙の注意事項に従
つて下の解答欄に記入すること。

問 題

		1	2	3	4	5	6	7	8	9	10	第9	(1)	1	2	3	4	5	6	7	8	9	10	第10	(1)	1	2	3	4	5	6	7	8	9	10												
第2	(1)	1	2	3	4	5	6	7	8	9	10	第9	(1)	1	2	3	4	5	6	7	8	9	10	第10	(5)	1	2	3	4	5	6	7	8	9	10												
(2)	1	2	3	4	5	6	7	8	9	10	(2)	1	2	3	4	5	6	7	8	9	10	(6)	1	2	3	4	5	6	7	8	9	10	第11	(1)	1	2	3	4	5	6	7	8	9	10			
(3)	1	2	3	4	5	6	7	8	9	10	(3)	1	2	3	4	5	6	7	8	9	10	(7)	1	2	3	4	5	6	7	8	9	10	第12	(1)	1	2	3	4	5	6	7	8	9	10			
(4)	1	2	3	4	5	6	7	8	9	10	(4)	1	2	3	4	5	6	7	8	9	10	(8)	1	2	3	4	5	6	7	8	9	10	第13	(1)	1	2	3	4	5	6	7	8	9	10			
(5)	1	2	3	4	5	6	7	8	9	10	(5)	1	2	3	4	5	6	7	8	9	10	(9)	1	2	3	4	5	6	7	8	9	10	第14	(1)	1	2	3	4	5	6	7	8	9	10			
(6)	1	2	3	4	5	6	7	8	9	10	(6)	1	2	3	4	5	6	7	8	9	10	(10)	1	2	3	4	5	6	7	8	9	10	第15	(1)	1	2	3	4	5	6	7	8	9	10			
(7)	1	2	3	4	5	6	7	8	9	10	(7)	1	2	3	4	5	6	7	8	9	10	第16	(1)	1	2	3	4	5	6	7	8	9	10	第16	(1)	1	2	3	4	5	6	7	8	9	10		
(8)	1	2	3	4	5	6	7	8	9	10	(8)	1	2	3	4	5	6	7	8	9	10	第17	(1)	1	2	3	4	5	6	7	8	9	10	第17	(1)	1	2	3	4	5	6	7	8	9	10		
(9)	1	2	3	4	5	6	7	8	9	10	(9)	1	2	3	4	5	6	7	8	9	10	第18	(1)	1	2	3	4	5	6	7	8	9	10	第18	(1)	1	2	3	4	5	6	7	8	9	10		
(10)	1	2	3	4	5	6	7	8	9	10	(10)	1	2	3	4	5	6	7	8	9	10	第19	(1)	1	2	3	4	5	6	7	8	9	10	第19	(1)	1	2	3	4	5	6	7	8	9	10		
第3	(1)	1	2	3	4	5	6	7	8	9	10	第11	(1)	1	2	3	4	5	6	7	8	9	10	第20	(1)	1	2	3	4	5	6	7	8	9	10	第20	(1)	1	2	3	4	5	6	7	8	9	10
第4	(1)	1	2	3	4	5	6	7	8	9	10	(2)	1	2	3	4	5	6	7	8	9	10	第21	(1)	1	2	3	4	5	6	7	8	9	10	第21	(1)	1	2	3	4	5	6	7	8	9	10	
(2)	1	2	3	4	5	6	7	8	9	10	(3)	1	2	3	4	5	6	7	8	9	10	第22	(1)	1	2	3	4	5	6	7	8	9	10	第22	(1)	1	2	3	4	5	6	7	8	9	10		
(3)	1	2	3	4	5	6	7	8	9	10	(4)	1	2	3	4	5	6	7	8	9	10	第23	(1)	1	2	3	4	5	6	7	8	9	10	第23	(1)	1	2	3	4	5	6	7	8	9	10		
(4)	1	2	3	4	5	6	7	8	9	10	第12	(1)	1	2	3	4	5	6	7	8	9	10	第24	(1)	1	2	3	4	5	6	7	8	9	10	第24	(1)	1	2	3	4	5	6	7	8	9	10	
(5)	1	2	3	4	5	6	7	8	9	10	第13	(1)	1	2	3	4	5	6	7	8	9	10	第25	(1)	1	2	3	4	5	6	7	8	9	10	第25	(1)	1	2	3	4	5	6	7	8	9	10	
(6)	1	2	3	4	5	6	7	8	9	10	第14	(1)	1	2	3	4	5	6	7	8	9	10	第26	(1)	1	2	3	4	5	6	7	8	9	10	第26	(1)	1	2	3	4	5	6	7	8	9	10	
(7)	1	2	3	4	5	6	7	8	9	10	(2)	1	2	3	4	5	6	7	8	9	10	第27	(1)	1	2	3	4	5	6	7	8	9	10	第27	(1)	1	2	3	4	5	6	7	8	9	10		
(8)	1	2	3	4	5	6	7	8	9	10	(3)	1	2	3	4	5	6	7	8	9	10	第28	(1)	1	2	3	4	5	6	7	8	9	10	第28	(1)	1	2	3	4	5	6	7	8	9	10		
(9)	1	2	3	4	5	6	7	8	9	10	(4)	1	2	3	4	5	6	7	8	9	10	第29	(1)	1	2	3	4	5	6	7	8	9	10	第29	(1)	1	2	3	4	5	6	7	8	9	10		
(10)	1	2	3	4	5	6	7	8	9	10	第15	(1)	1	2	3	4	5	6	7	8	9	10	第30	(1)	1	2	3	4	5	6	7	8	9	10	第30	(1)	1	2	3	4	5	6	7	8	9	10	
第5	(1)	1	2	3	4	5	6	7	8	9	10	(2)	1	2	3	4	5	6	7	8	9	10	第31	(1)	1	2	3	4	5	6	7	8	9	10	第31	(1)	1	2	3	4	5	6	7	8	9	10	
(2)	1	2	3	4	5	6	7	8	9	10	(3)	1	2	3	4	5	6	7	8	9	10	第32	(1)	1	2	3	4	5	6	7	8	9	10	第32	(1)	1	2	3	4	5	6	7	8	9	10		
(3)	1	2	3	4	5	6	7	8	9	10	(4)	1	2	3	4	5	6	7	8	9	10	第33	(1)	1	2	3	4	5	6	7	8	9	10	第33	(1)	1	2	3	4	5	6	7	8	9	10		
(4)	1	2	3	4	5	6	7	8	9	10	(5)	1	2	3	4	5	6	7	8	9	10	第34	(1)	1	2	3	4	5	6	7	8	9	10	第34	(1)	1	2	3	4	5	6	7	8	9	10		
(5)	1	2	3	4	5	6	7	8	9	10	(6)	1	2	3	4	5	6	7	8	9	10	第35	(1)	1	2	3	4	5	6	7	8	9	10	第35	(1)	1	2	3	4	5	6	7	8	9	10		
第6	1	2	3	4	5	6	7	8	9	10	(3)	1	2	3	4	5	6	7	8	9	10	第36	(1)	1	2	3	4	5	6	7	8	9	10	第36	(1)	1	2	3	4	5	6	7	8	9	10		
第7	(1)	1	2	3	4	5	6	7	8	9	10	(4)	1	2	3	4	5	6	7	8	9	10	第37	(1)	1	2	3	4	5	6	7	8	9	10	第37	(1)	1	2	3	4	5	6	7	8	9	10	
(2)	1	2	3	4	5	6	7	8	9	10	第16	(1)	1	2	3	4	5	6	7	8	9	10	第38	(1)	1	2	3	4	5	6	7	8	9	10	第38	(1)	1	2	3	4	5	6	7	8	9	10	
(3)	1	2	3	4	5	6	7	8	9	10	第17	(1)	1	2	3	4	5	6	7	8	9	10	第39	(1)	1	2	3	4	5	6	7	8	9	10	第39	(1)	1	2	3	4	5	6	7	8	9	10	
(4)	1	2	3	4	5	6	7	8	9	10	第18	(1)	1	2	3	4	5	6	7	8	9	10	第40	(1)	1	2	3	4	5	6	7	8	9	10	第40	(1)	1	2	3	4	5	6	7	8	9	10	
(5)	1	2	3	4	5	6	7	8	9	10	第19	(1)	1	2	3	4	5	6	7	8	9	10	第41	(1)	1	2	3	4	5	6	7	8	9	10	第41	(1)	1	2	3	4	5	6	7	8	9	10	
第8	1	2	3	4	5	6	7	8	9	10	(2)	1	2	3	4	5	6	7	8	9	10	第42	(1)	1	2	3	4	5	6	7	8	9	10	第42	(1)	1	2	3	4	5	6	7	8	9	10		

問題

CP28	(1)	2	3	4	5	6	7	8	9	10	CP31	(1)	2	3	4	5	6	7	8	9	10	CP32	(1)	2	3	4	5	6	7	8	9	10	
	(2)	1	2	3	4	5	6	7	8	9	CP31	(2)	1	2	3	4	5	6	7	8	9	CP32	(2)	1	2	3	4	5	6	7	8	9	10
	(3)	1	2	3	4	5	6	7	8	9	CP31	(3)	1	2	3	4	5	6	7	8	9	CP32	(3)	1	2	3	4	5	6	7	8	9	10
	(4)	1	2	3	4	5	6	7	8	9	CP31	(4)	1	2	3	4	5	6	7	8	9	CP32	(4)	1	2	3	4	5	6	7	8	9	10
	(5)	1	2	3	4	5	6	7	8	9	CP31	(5)	1	2	3	4	5	6	7	8	9	CP32	(5)	1	2	3	4	5	6	7	8	9	10
	(6)	1	2	3	4	5	6	7	8	9	CP31	(6)	1	2	3	4	5	6	7	8	9	CP32	(6)	1	2	3	4	5	6	7	8	9	10
	(7)	1	2	3	4	5	6	7	8	9	CP31	(7)	1	2	3	4	5	6	7	8	9	CP32	(7)	1	2	3	4	5	6	7	8	9	10
	(8)	1	2	3	4	5	6	7	8	9	CP31	(8)	1	2	3	4	5	6	7	8	9	CP32	(8)	1	2	3	4	5	6	7	8	9	10
	(9)	1	2	3	4	5	6	7	8	9	CP31	(9)	1	2	3	4	5	6	7	8	9	CP32	(9)	1	2	3	4	5	6	7	8	9	10
	(10)	1	2	3	4	5	6	7	8	9	CP31	(10)	1	2	3	4	5	6	7	8	9	CP32	(10)	1	2	3	4	5	6	7	8	9	10
CP30	(1)	2	3	4	5	6	7	8	9	10	CP33	(1)	2	3	4	5	6	7	8	9	10	CP34	(1)	2	3	4	5	6	7	8	9	10	
	(2)	1	2	3	4	5	6	7	8	9	CP33	(2)	1	2	3	4	5	6	7	8	9	CP34	(2)	1	2	3	4	5	6	7	8	9	10
	(3)	1	2	3	4	5	6	7	8	9	CP33	(3)	1	2	3	4	5	6	7	8	9	CP34	(3)	1	2	3	4	5	6	7	8	9	10
	(4)	1	2	3	4	5	6	7	8	9	CP33	(4)	1	2	3	4	5	6	7	8	9	CP34	(4)	1	2	3	4	5	6	7	8	9	10
	(5)	1	2	3	4	5	6	7	8	9	CP33	(5)	1	2	3	4	5	6	7	8	9	CP34	(5)	1	2	3	4	5	6	7	8	9	10
	(6)	1	2	3	4	5	6	7	8	9	CP33	(6)	1	2	3	4	5	6	7	8	9	CP34	(6)	1	2	3	4	5	6	7	8	9	10
	(7)	1	2	3	4	5	6	7	8	9	CP33	(7)	1	2	3	4	5	6	7	8	9	CP34	(7)	1	2	3	4	5	6	7	8	9	10
	(8)	1	2	3	4	5	6	7	8	9	CP33	(8)	1	2	3	4	5	6	7	8	9	CP34	(8)	1	2	3	4	5	6	7	8	9	10
	(9)	1	2	3	4	5	6	7	8	9	CP33	(9)	1	2	3	4	5	6	7	8	9	CP34	(9)	1	2	3	4	5	6	7	8	9	10
	(10)	1	2	3	4	5	6	7	8	9	CP33	(10)	1	2	3	4	5	6	7	8	9	CP34	(10)	1	2	3	4	5	6	7	8	9	10
CP31	(1)	2	3	4	5	6	7	8	9	10	CP35	(1)	2	3	4	5	6	7	8	9	10	CP36	(1)	2	3	4	5	6	7	8	9	10	
	(2)	1	2	3	4	5	6	7	8	9	CP35	(2)	1	2	3	4	5	6	7	8	9	CP36	(2)	1	2	3	4	5	6	7	8	9	10
	(3)	1	2	3	4	5	6	7	8	9	CP35	(3)	1	2	3	4	5	6	7	8	9	CP36	(3)	1	2	3	4	5	6	7	8	9	10
	(4)	1	2	3	4	5	6	7	8	9	CP35	(4)	1	2	3	4	5	6	7	8	9	CP36	(4)	1	2	3	4	5	6	7	8	9	10
	(5)	1	2	3	4	5	6	7	8	9	CP35	(5)	1	2	3	4	5	6	7	8	9	CP36	(5)	1	2	3	4	5	6	7	8	9	10
	(6)	1	2	3	4	5	6	7	8	9	CP35	(6)	1	2	3	4	5	6	7	8	9	CP36	(6)	1	2	3	4	5	6	7	8	9	10
	(7)	1	2	3	4	5	6	7	8	9	CP35	(7)	1	2	3	4	5	6	7	8	9	CP36	(7)	1	2	3	4	5	6	7	8	9	10
	(8)	1	2	3	4	5	6	7	8	9	CP35	(8)	1	2	3	4	5	6	7	8	9	CP36	(8)	1	2	3	4	5	6	7	8	9	10
	(9)	1	2	3	4	5	6	7	8	9	CP35	(9)	1	2	3	4	5	6	7	8	9	CP36	(9)	1	2	3	4	5	6	7	8	9	10
	(10)	1	2	3	4	5	6	7	8	9	CP35	(10)	1	2	3	4	5	6	7	8	9	CP36	(10)	1	2	3	4	5	6	7	8	9	10

(8) 磁気インク文字(magnetic ink character)

磁気インク文字については次の説明をする。

(a) 磁気インク文字は、磁気インク文字読取装置(magnetic ink character reader, 略してMICR)によって読込まれる。

(b) MICRのコード

(c) MICRを国際的なコードにするために、磁気印字のための標準的な書式が考えられている。

(d) 特殊なインクを使用するので、普通のタイプライタなどでは印字できない。MICR専用のものが必要である。

(9) 磁気テープ

磁気テープのコードと、データの記録方法を中心に説明する。

(a) コード

データは、7トラックと9トラックのコードで表現される。

次のような図を示すとよい。

9トラックの場合

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	&	.	-	*	/	'	%	@		
b3																																														
b1																																														
b5																																														
bP																																														
b6																																														
b7																																														
b8																																														
b2																																														
b4																																														

1インチに記録できるデータは100ビットから、多いものでは3,000ビット以上である。

奇偶検査の機能に触れ、奇数パリティ (odd parity) と偶数パリティ (even parity) の説明をする。

ブロック間間隙 (inter-block gap) について説明する。

(b) データの記録

かつては、磁気テープにデータを記録するには、変換装置を使用して、他の媒体から移さなければならなかったが、最近では直接、磁気テープにデータを記録することができる。

磁気テープにデータを記録する長所として、次のようなことを説明する。

- (i) データを記憶装置に読込んで、磁気テープに書出すまでの間に、そのデータの変更が可能である。
- (ii) 磁気テープ内のある場所に入っている特定のデータの変更が容易である。
- (iii) 入出力の速度が速い。

3.5 データの変換

今までに多くの記録媒体の説明をしてきたが、これらはどの機械でも使えるとは限らない。ある機械でその記録媒体のデータを読み込んで処理ができて、他の機械ではできなかったりする。そこで、データ媒体間での変換が必要になる。データ変換用として開発されている機械について説明する。

次のような媒体変換について述べる。

紙テープから	{ 紙カード 磁気テープ	プリント・パンチ ・タグから	{ 紙テープ 紙カード
磁気テープから	{ 紙テープ 紙カード	パーフォレート・ クーポンから	{ 紙カード 紙テープ 磁気テープ
紙カードから	{ 紙テープ 磁気テープ	エッジ・ノッチ・ カードから	{ 紙カード 紙テープ

指導上の留意点

- (1) この章で述べているデータ収集では、データ伝送については取扱わない。
- (2) ここでは、主としてコンピュータに入力するデータを記録する媒体や、コードについて説明する。
説明にあたっては、できるだけ実物を見せた方がよい。
- (3) その他の媒体（光学文字、磁気インクなど）は、紹介程度にとどめてもよい。その場合についても、できるだけ写真などの資料の準備をしたい。
- (4) コードをはじめに考案したのはモールスであるが、紙テープや紙カードには、独特なコードの使い方があることを、例を示して理解させる。
- (5) 全体として、この章は単なる説明の羅列に終ると、生徒の興味を減退させるから、技術的に立入って説明することは避けて、データ収集の媒体についての考え方を理解させるということにねらいをしばった方がよい。詳しいことは、第2部で説明する。

参考文献

1. MODERN DATA PROCESSING

Robert R. Arnold

Harold G. Hill

Aylmer V. Nichols

John Wiley & Sons, INC.

1969 3,580円

このカリキュラム作成の基本文献である。

2. データ・プロセッシング

加藤録吉 著

日本経営出版会

昭和43.7 900円

データ処理に関する基礎的な概念について触れている。

3. 電子計算機とデータ処理

鶴沢昌和 訳

東洋経済新聞社

昭和44.9 1,600円

第2章 計算の歴史のハイライトがデータ処理の歴史を知る上で、参考になる。

4. 事務機械化の進め方

小沢暢夫 著

日刊工業新聞社

昭和43.9 740円

事務機械や、事務処理過程を知る上で役に立つ。

5. 事務の機械化

寺島正朝

著
篠原靖市

産学社

昭和42.12 750円

第1部 事務の機械化、第2部 データ処理の機械化が参考になる。

6. 経 営

高宮 晋
野田一夫 著
有斐閣

昭39.7 480円

現代における企業の構造と機能を知る上で役立つ。

7. 電子計算機概論

— 計算機システムと情報処理 —

池田敏雄 著
オーム社

昭45 1,000円

第1章 電子計算機の発展過程と、第4章 周辺装置が参考になる。

8. 電子計算機 最新プログラミング

竹下 亨 著
日本経営出版会

昭43.5 2,400円

第1部 電子計算機の基本知識と第IX部 データ処理における工程管理が参考になる。

第2部 コンピュータのハードウェア

教育の目標

1939年、ハーバード大学のDr.Howard Aikenが発表した計算機開発計画は、自動計算機から電子計算機へ発展する最後のステップであったといわれる。

Aikenの計算機は、1944年に完成したが、その後コンピュータは急速に進歩した。

第2部の目標は、こうして急速に進歩し、情報処理の新しい重要な武器となったコンピュータについて、その概要を理解させることである。ハードウェアの機能と構成についての知識は、コンピュータ理解の基礎となるので、プログラミング技法やプログラミング・システム、コンピュータの操作などを教える前にコンピュータのハードウェアについての理解を深めておくことが必要である。

内容のあらまし

内 容	時間数
ハードウェアのあらまし	
コンピュータの歴史	4
現代のコンピュータ	3
コンピュータ・システム	3
入出力装置	
入出力装置の機能	6
各種の入出力装置	8
記憶装置	
データの表現	4
内部記憶（記憶原理と各種記憶）	4
補助記憶	3
外部記憶	2
演算・制御装置	
演算の原理	4
中央演算処理装置の基本要素	2
制御機構	3
算術論理演算機構	4
合 計	50

第1章 ハードウェアのあらまし

この章では次の用語を教える。

アナログ型コンピュータ、デジタル型コンピュータ（計数型計算機）、集積回路、専用コンピュータ、超小型コンピュータ、トランジスタ、ハイブリッド型コンピュータ、汎用コンピュータ、プロセス制御、自動計算機、大規模集積回路（LSI）、プログラム内蔵式、プログラム、ハードウェア、ソフトウェア、精度

目 標

コンピュータ進歩の過程を歴史的にたどる。コンピュータの進歩は、他の科学技術分野の発展に依存する面が大きいので、電子工学など関連する分野の発展がコンピュータに与えた影響についても触れる。コンピュータの発展の過程でコンピュータにどのような期待がかけられ、それがどう実現されてきたかを述べる。コンピュータが計算の道具から、情報処理の武器へと発展し、人工頭脳と呼ばれるまでになった技術的進歩と、利用分野の広がりについて説明する。さらに第2章以下の導入として、ハードウェアについて概括的に説明する。

内 容

1.1 コンピュータの歴史

コンピュータの歴史は、1946年に真空管を使った最初の実用機ENIAC（Electro-nic Numerical Integrator And Calculator）が登場してから25年にしかならないが、計算する機械の歴史は、そろばんの歴史にまでさかのぼることができる。この節では、コンピュータが開発されるまでの歴史を述べる。第1部第2章の復習を兼ねる。

本格的なコンピュータが登場するまでの歴史については、次のような点を説明する。

- (1) そろばんの時代 17世紀にB. Pascalの加算器が出現するまで。
- (2) 計算する機械の登場 17世紀から18世紀にかけて次のような機械が開発された。
 - (a) J. Napierの乗算道具
 - (b) B. Pascalの加算器

(c) G.W. Leibnitzの乗算器

(3) コンピュータが登場するまでの間の自動計算の試み。

(a) Charles Babbageのdifference engineとanalytical engine

現在のコンピュータがBabbageの自動計算機の原理とよく似ていることを述べる。

(b) Herman Hollerithのパンチ・カード式統計機械

(c) Howard Aikenの電気機械式計算機 Mark-I

(4) ENIACの登場以降

(a) ENIACによってはじめて機械的要素が電子的要素に置き換えられた。

(b) 商用の汎用コンピュータUNIVACの開発

(c) John Von Neumannの2進数表示とプログラム内蔵式の理論

(d) EDSAC(Electronic Delayed Storage Automatic Computer)

の開発、プログラム内蔵式が取入れられた。

本格的なコンピュータが登場してからの発展の歴史については次の点を説明する。

(1) 真空管(演算制御装置)と水銀遅延線(記憶装置)の使用

(2) トランジスタ、ダイオードと磁気コアの使用

(3) コンピュータの互換性や拡張性、安定性などの拡大

(4) 集積回路(IC)、大規模集積回路(LSI)の出現、記憶装置の増大、演算処理速度の増

加、小型化の実現

(5) 多種多様な入出力装置の登場

1.2 現代のコンピュータ

現在、コンピュータがどのような分野で使われているかを簡単に説明し、コンピュータを必要とする分野の作業の性質や、作業量などにより、いろいろな機能や規模のコンピュータが必要なことを説明する。

現代広く使われているコンピュータを次のように分類して説明する。

(1) 目的別の分類

(a) 専用コンピュータ

軍船用潜水艦の航海制御、飛行機やミサイル制御用コンピュータ、石油精製、化学工場、製鉄、発電などで生産の自動化とともに、プロセス制御用として使われるようになったコンピュータの例などをあげる。

(b) 汎用コンピュータ

広い分野の業務や、作業を処理することができるコンピュータで、給与計算、統計業務、販売分析、請求書の処理、原価計算、人員配置、生産計画、在庫管理、技術計算などに広く使われることを説明する。

(2) 能力別の分類

コンピュータの初期の段階では、能力はその物理的大きさで測られた。しかし電子工学の進歩とともに、コンピュータの小型化、高性能化が進み、コンピュータの物理的大きさは能力を表わす尺度にならなくなった。そこで物理的な大きさに代って主記憶装置の容量や、コンピュータの価格が能力の尺度として使われるようになったことを説明する。

コンピュータを次のように分けて説明する。

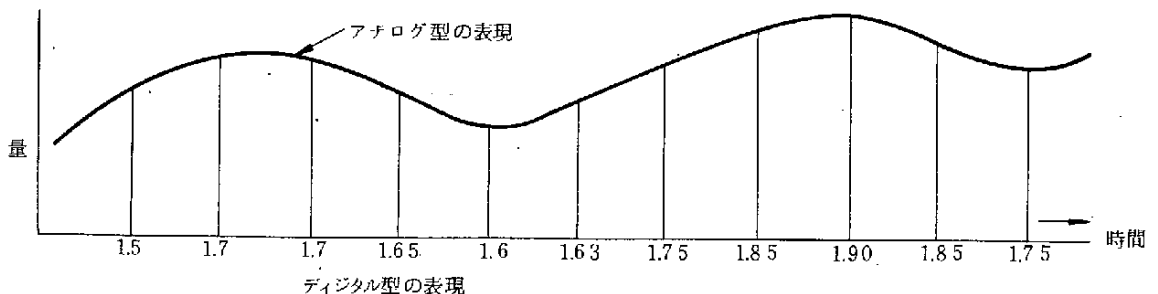
- (a) 大型コンピュータ 買取り価格 250,000千円以上
- (b) 中型コンピュータ " 40,000千円以上
- (c) 小型コンピュータ " 10,000千円以上
- (d) 超小型コンピュータ " 10,000千円未満
- (e) ミニ・コンピュータ 主記憶が2,000ビット以上で、中央演算処理装置の価格が5,000千円以下

それぞれのクラスのコンピュータの例をあげて、賃借料、機能、性能の比較、使用例などを説明するとよい。

(3) データの取扱い方による分類

コンピュータでデータを取扱う方法にアナログ型とデジタル型がある。

次の図を示して、アナログ型とデジタル型の違いを説明する。



データを取扱う方法によって、次のようにコンピュータを分けることができる。

(a) アナログ型コンピュータ

(b) デジタル型コンピュータ

(c) ハイブリッド型コンピュータ

アナログ型コンピュータは、石油精製、製紙、製鉄などの工場のプロセス制御用コンピュータとして使われている。一般に広く使われているのは、デジタル型コンピュータである。デジタル型コンピュータの特徴について次の点をあげることができる。

- (a) 数字で表わされている実数や文字を扱う。
- (b) 精度の高い加減乗除、比較演算を行なうことができる。
- (c) 文字や数字を正確に処理できる。

ハイブリッド型コンピュータでは、アナログ型とデジタル型の両方のデータを扱う。そのためにアナログ型とデジタル型相互のデータ変換装置が必要である。

以上の説明をしたあとで、初級カリキュラムではデジタル型のコンピュータだけを扱うことを明確にしておく必要がある。

1.3 コンピュータ・システム

次の点についてそれぞれ説明する。

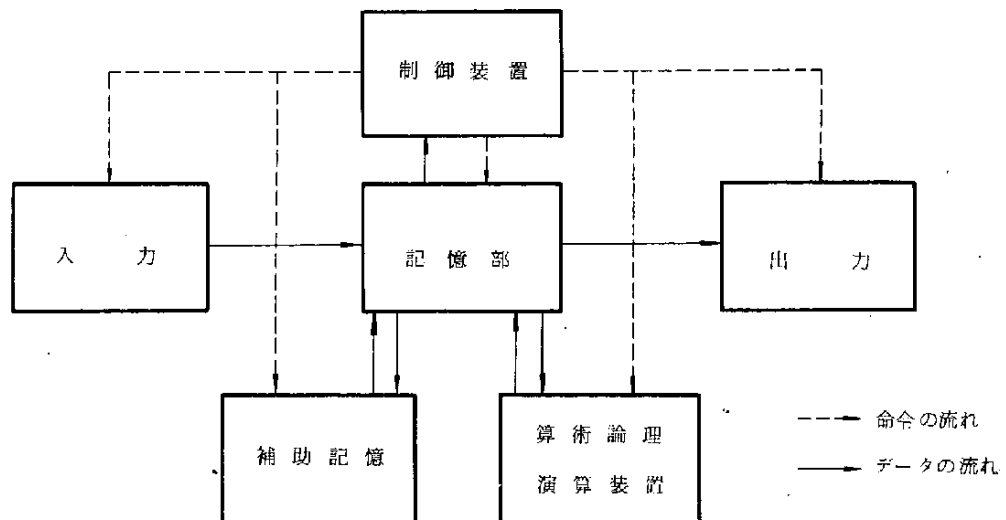
コンピュータの構成

コンピュータの働き

ハードウェアとソフトウェア

(1) コンピュータの構成

次のような図を示してコンピュータを構成している主な装置と各装置の役割を簡単に説明する。



(2) コンピュータの働き

コンピュータの働きを理解させるために、上記の図を使って、次のような順序で処理の過程を簡単に説明する。

- (a) 処理するデータと処理に必要なプログラムを紙テープ、紙カード、磁気テープなどの媒体に記録する。
- (b) プログラムとデータを入力装置から読み、記憶装置に記憶する。
- (c) 命令を制御装置で解読して実行する。
- (d) 命令を実行するとデータが記憶装置から、演算装置に移され処理される。
- (e) 処理されたデータが、記憶装置に戻される。
- (f) 処理されたデータが記憶装置から出力されて紙テープ、紙カード、磁気テープ、印刷用紙などの媒体に記録される。

(3) ハードウェアとソフトウェア

ソフトウェアということばが生まれた理由を述べ、ハードウェアとソフトウェアの概念を簡単に説明する。この段階では、まだソフトウェアの概念を正確に理解させることはできないので、ことばになじませる程度でよい。

指導上の留意点

この章は、コンピュータの説明の導入部分であるから、説明が詳しすぎるとかえって理解をさまたげることになる。できるだけコンピュータを概括的に掌握させ、コンピュータの意味や役割について正しい理解を持たせるようにする。コンピュータの発展の過程をたどったり、コンピュータがどのように使われているかという面からながめてみることは、コンピュータの役割を理解する助けになるであろう。

コンピュータの能力別分類には、いろいろな区分の仕方があるが、日本電子計算機(株)(J E C C)の分類を基準にするのがよいであろう。

第2章 入出力装置

この章では次の用語を教える。

オフライン、オンライン、カード送り、緩衝域(バッファ)、入出力装置、入出力チャネル、カード読取せん孔装置、紙テープせん孔装置、磁気テープ装置、ミリ秒、ファイル保護リング、物理レコード、記録密度、文字セット、磁気インク文字読取装置、出力装置、消去、垂直検査、水平検査、スペース、入出力制御装置、制御卓、制御盤、妥当性検査、端末装置、入力装置、バックスペース、印刷装置、スキップ、キャリッジ・テープ、マイクロ・フィルム・システム、光学文字読取装置(OOR)、映像表示装置、ライト・ペン、音声応答システム、ターンアラウンド、ブラシ、ブロック、ヘッド、ライン・プリンタ(行印字機)、レコード間間隙、中央演算処理装置

目 標

コンピュータを構成する装置の1つである入出力装置について、次のような点を理解させる。

- (1) 人とコンピュータとの交流は、入出力装置を介して行なわれる。
- (2) データは、入出力媒体を介してコンピュータに入出力される。
- (3) 入出力装置の種類と入出力の仕組み。

内 容

入力装置、出力装置の機能、代表的な入力出力の種類とその役割について説明する。

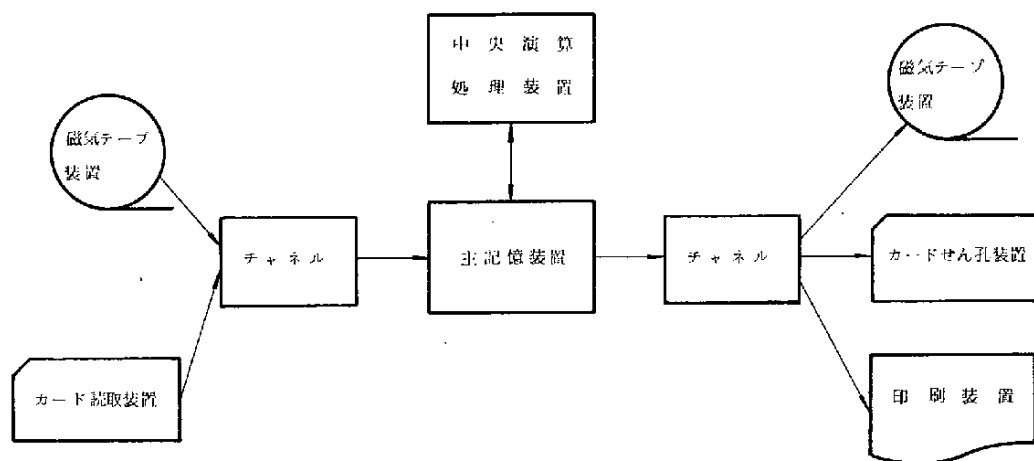
2.1 入出力装置の機能

入出力装置がデータを入出力する速度は、コンピュータの中央演算処理装置に比較して非常に遅いので(速いものでもミリ秒単位)、両者の速度を調整し処理を円滑に行なうために、入出力装置に次のような機能が必要である。

- (1) 入出力チャネルの機能
- (2) 緩衝域(バッファ)の機能

以上のそれぞれの機能を説明し、この2つの機能によってデータの入出力と中央演算処理装置におけるデータの処理とを独立して並行的に行なうことができ、高速の演算処理が可能になることを説明する。

次の図を用いるとよい。第1章のコンピュータの構成図と対応させて、中央演算処理装置の概念を導入する。中央演算処理装置は、制御装置と算術論理演算装置に対する総称である。



チャンネルをデータが通過するときには、次の2つの方法があることを説明する。

- (1) 低速の入出力装置のときは、チャンネルの通過路を入出力装置が共用し、同時に複数の入出力装置のデータを通過させる。
- (2) 高速の入出力装置のときは、一時には1つの入出力装置だけがチャンネルの通過路で占有してデータを通過させる。

2.2 各種の入出力装置

入出力装置は、コンピュータの利用分野の広がりとともに人とコンピュータの交流を容易にするという目的のもとに数多くの種類が開発された。

その中から、代表的なものを取り上げて説明する。

(1) カード読取せん孔装置

紙カードによる入出力は、コンピュータの入出力として最も一般的であることを述べ、次のよ

うな点を説明する。

(a) 紙カードの利点と欠点

せん孔、検孔、分類などが自由にできる。

1つのデータが80けたに制限されている。

入出力速度が遅い。

(b) 紙カードの読取り方式

ブラシ接触式(機械式)と光電読取り式(電子式)とがある。

読取りを示す図を用いて説明するとよい。カード送りについて説明する。

(c) 紙カードのせん孔方式

(d) 読取り速度とせん孔速度

(e) 低速である。

非常に低速なので、コンピュータの性能を生かすために同時に複数個の作業を行なうなどの方法を考える必要がある。

(2) 紙テープ読取装置、紙テープせん孔装置

紙テープ読取装置とせん孔装置は、紙テープを媒体として入出力を行なう。入出力の原理は、カード読取せん孔装置の原理とよく似ている。

以下のことを説明する。

(a) 紙カードと比較した場合の利点と欠点

データの長さを一定にする必要がない。

コンピュータだけでなく、いろいろなデータ処理機械で共通して使える。

データの追加や削除がむずかしい。

分類や照合ができない。

(b) 読取り速度とせん孔速度

(3) 磁気テープ装置

次のような点について説明する。

(a) 磁気テープ装置の特徴と機能

磁気テープの実物を見せるとよい。

磁気テープは、入出力装置としてだけでなく、補助記憶装置としても重要である。

磁気テープは高速であり、大容量のデータを記録できる。

分類を行なうときの重要な媒体である。

磁気テープに記憶された内容は、直接目で見ることはいできない。

等速呼出しができない。

入出力の両方に使用できる。

(b) 磁気テープ装置の機構

(i) ファイル・リールと巻取りリール

(ii) 読取りヘッドと書込みヘッド

(iii) バックスペースと消去(erase)

(iv) 真空コラム機構

(v) ファイル保護リングの働き

(c) 磁気テープの記録形式

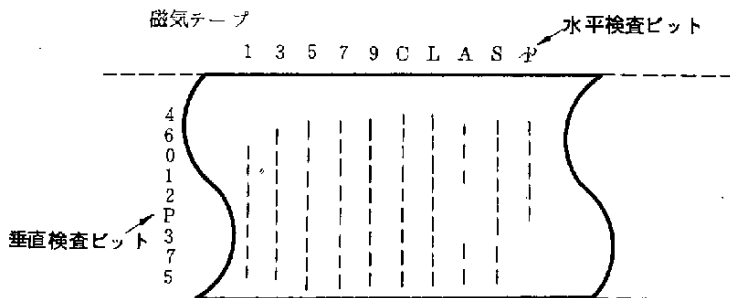
(i) 磁気テープの始めと磁気テープの終り

(ii) ブロック(物理レコード)とブロック間間隙(レコード間間隙)

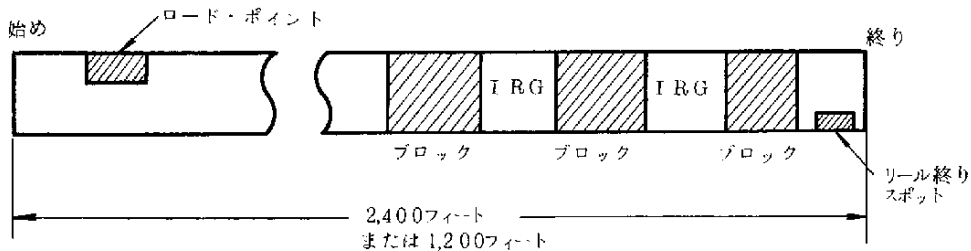
(iii) 記録密度

次のような図を書いて説明するとよい。

データの記録表現



データの記録形式



(d) 読取り、書込み速度

磁気テープの読取り速度と書込み速度は同じである。

読み書き速度は、磁気テープの物理的な速度と記憶密度によって決る。 する際、

磁気テープの読み書き速度は15K~180K字/秒の範囲である。

(4) 印刷装置

印刷装置について、次のことを説明する。

(a) ライン・プリンタ (Line Printer)

印字機構には {
ドラム式ライン・プリンタ
チェイン式ライン・プリンタ

がある。

各印字機構の簡単な図を示して、印刷の仕組みと1行の最大印刷字数、用紙のスペースやスキップの制御、キャリッジ・テープの役割などについて説明する。プリント・バッファについても触れておくことが望ましい。

(b) 印刷装置の速度

印刷速度、スキップとスペースの速度。印刷装置の活字の数(文字セット)によって速度が異なる。

(c) その他の方式の印刷装置

写真植字による印刷方式について、特徴を簡単に説明する。

(5) 制御卓

制御卓は、制御盤とタイプライタから成っており、主としてコンピュータ・システムを監視し、システムに必要な指示を与えるために使われる。次のようなことを説明する。

(a) 制御盤の機能

コンピュータの内部の状態を表わす。

(b) タイプライタの役割

メッセージの印刷、コンピュータに対する必要な指示の入力

(c) 操作員の操作

処理の開始と終止、エラー・リセット、入出力装置の選択、データのマニュアル入力など

(6) マイクロ・フィルム・システム

次のような点をあげて、マイクロ・フィルム・システムの特徴を説明する。

(a) ライン・プリンタに比べてはるかに高速である。

(b) 印刷されたものがコンパクトである。

ライン・プリンタの1,600ページ分が手のひらにのる。

マイクロ・フィルムからコピーをとるために、リーダー・プリンタがある。

(7) 磁気インク文字読取装置(MICR)

次のことを説明する。

(a) ドキュメントに磁気インクを使って印刷された文字を磁気ヘッドで直接読取る装置である。

(b) ドキュメントを分類する機能をもつ。

(c) オンラインでもオフラインでも使える。

(d) ドキュメントに1行印刷された文字を毎分750～1,600枚で読む。

(e) 銀行の小切手や債券の処理に広く使われている。

(8) 光学文字読取装置(OCR)

光学マーク読取装置(OMR)

次の点を説明する。

(a) ドキュメントに印刷された文字を光学的に直接読取る装置である。

(b) ドキュメントとしては、通常用紙、カード、ジャーナル・テープなどが使える。

(c) マーク、印刷された数字、文字、特殊文字、手書き文字などを読むことができる。

(d) 料金徴収事務などでターンアラウンド方式と呼ばれる処理に使われている。

(9) タイプライタ型端末装置

オンライン・システムのタイプライタ型の入出力装置について簡単に説明する。銀行業務など特定の用途向きに作られた端末機と汎用端末機があることを説明する。

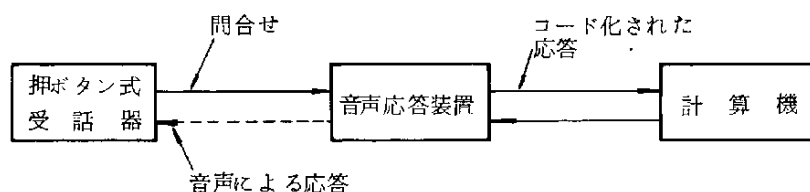
(10) 映像表示装置

映像表示装置の用途と特徴について説明する。

ライト・ペンについても簡単に触れるとよい。

(11) 音声応答システム

音声応答システムの用途と特徴について説明する。次のような図を用いるとよい。



指導上の留意点

入出力装置については、できるだけ実物を見せて説明することが有効である。実物を見せることがむずかしければ、スライドを使ってもよい。入出力装置は、コンピュータ・システムへのデータの入出力をできるだけ容易にし、かつスピード・アップするという目的で、利用分野に応じた開発がなされてきた。したがって特定目的の入出力装置については、それがどのような分野で、どのように使われているかを実例を示して説明する必要がある。

入出力装置を説明する際、入出力装置に共通したことがらとして次のような点を説明する必要がある。

(1) 入出力媒体上での数字や文字の表現方式と標準化への問題

標準化については、ISO規格やJIS規格があるものは、それらを紹介する。

(2) 入出力装置の制御

プログラムによる制御と機械的な制御

オンライン制御とオフライン制御

(3) 中央演算処理装置と入出力装置の間の速度調整やコード変換 入出力制御装置の働きを簡単に述べる。

(4) 入出力誤りの検査

水平検査と垂直検査、妥当性検査、比較検査など

(5) 入出力制御装置の役割

時間に余裕があれば、オンライン・システムについて簡単に説明してもよい。端末装置、回線、モデム（変復調装置）、回線制御装置、中央演算処理装置などの働きを簡単に説明する。

第3章 記 憶 装 置

この章では次の用語を教える。

アクセス、アクセス・アーム、アクセス時間、アドレス(番地)、拡張2進化10進コード(EBCDIC)、下位けた(単位けた)、記憶、記憶容量、基準アドレス(ベース・アドレス)、基準レジスタ、検査ビット、語、語長、固定記憶、再配置可能、字(キャラクタ)、磁気コア記憶、磁気ディスク記憶、磁気ドラム記憶、磁気薄膜、磁気フィルム記憶、磁気カード記憶、磁気テープ記憶、磁気カード、順次アクセス、順次処理、上位けた シリンダ、セクタ、絶対アドレス、相対アドレス、直接アクセス、ディスク・パック、等速呼出し(ランダム・アクセス)、等速呼出し記憶、トラック、ナノ秒、2進数、バイト、ビット、マイクロ秒、パック、アンパック、記憶装置、主記憶、外部記憶、補助記憶、2進化10進コード、パリティ・ビット、ゾーン・ビット、USASCIIコード、シーク時間、回転待ち時間、データ転送時間、ワード・マシン、スイッチ、10進数、サイクル

目 標

データの記憶は、データ処理の中で大変重要な部分である。入力媒体に記録されたデータは、入力装置で読込まれ、主記憶に記憶される。また必要に応じて、外部記憶や補助記憶に転送される。記憶装置には、それぞれ固有のデータ表現形式がある。この章では、データの表現方法、記憶の機能、記憶の原理などを理解させる。

内 容

3.1 データの表現

最初にデータが入力媒体に記録されたり、記憶装置に記憶されるときの状態を説明する。入力媒体や記憶装置の中では、ある状態が存在するかないかという2つの状態の組合せによってデータが表現されている。次のような例をあげてONの状態、OFFの状態を説明する。

- (1) 紙カード………せん孔されているか、されていないか
- (2) トランジスタ………電流が流れているか、流れていないか

- (3) 磁気……1方向に磁化されているか、他方向に磁化されているか
- (4) リレー、スイッチ……開いているか、閉じているか
- (5) 電圧……かかっているか、かかっていないか

コンピュータでは、ON, OFFの状態の組合せで数や文字を表現するので2進数表現が重要であることを述べる。2進数による数の表現について説明し、10進数と2進数の間の数の変換について簡単な練習問題をさせるとよい。数字や文字が記憶装置に記憶されるときは、2進数を基本にしたコードに変換される。一般に使われているコードとして、次のコードを説明する。それぞれのコードで数を表現したときの正、負の符号の表わし方についても触れる。

(1) 2進コード

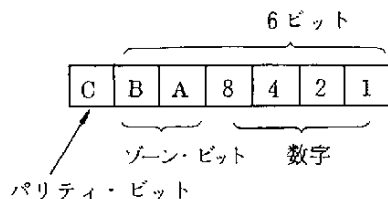
ビット(bit)と語(word)について説明する。

(2) 2進化10進コード

4ビットで数字を表現する。

(3) 標準2進化10進コード(Standard Binary Coded Decimal Interchange Code)

6ビットで数字と文字を表現する。次の図を示し、パリティ・ビットとゾーン・ビットの意味を説明する。



(4) 拡張2進化10進コード(Extended Binary Coded Decimal Interchange Code-EBCDIC)

8ビットで英数字を表現する。

バイト(byte)の意味を説明する。

ゾーン数とパック(pack)された数について説明する。

(5) USASCIIコード(U.S.A Standard Code For Information Interchange Code)

7ビットあるいは8ビット(USASCII-8)の2進数で表現する。

3.2 主記憶

内部記憶または1次記憶とも呼ばれ、コンピュータによって直接制御される。次の点を説明する。

- (1) ビットとバイト
- (2) 語単位 (ワード・マシン) と字 (キャラクタ) (character) 単位
1 語の長さ (語長) についても触れる。
- (3) 数字のパックとアンパック (unpack)
- (4) 固定長と可変長
- (5) アドレス (番地) (address)

アドレスについては次のような概念を説明する必要がある。

- (a) 基準レジスタ (ベース・レジスタ)
- (b) 基準アドレス (ベース・アドレス) とディスプレイメント
- (c) 絶対アドレス
- (d) 相対アドレス
- (e) 再配置可能

基準アドレスとディスプレイメントを使って表現したアドレスを相対アドレスといっている。
相対アドレスを使うと、プログラムの再配置が可能になる (relocatable)。

- (6) 内容を移動したときの状態、データの消える仕組み。
- (7) 固定記憶 (read only memory)
- (8) 補助記憶、外部記憶との関連

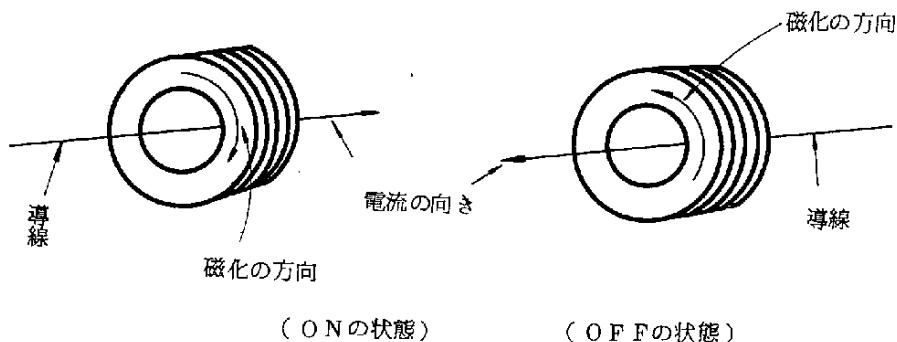
内部記憶の代表的なものとして、磁気コア記憶と、磁気フィルム記憶について説明する。

- (1) 磁気コア記憶

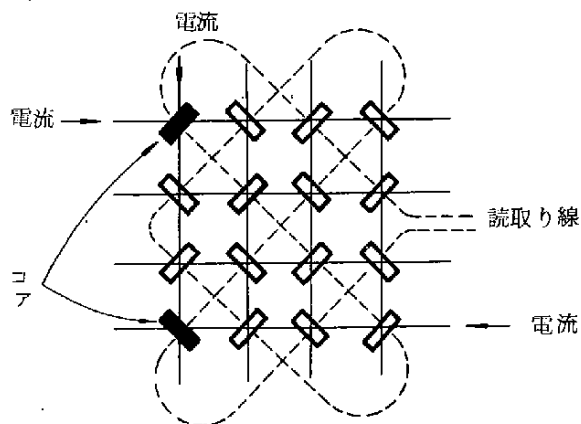
現在のコンピュータでは、ほとんどが主記憶に磁気コアを用いている。

磁気コアの仕組みについて、次の図で説明する。

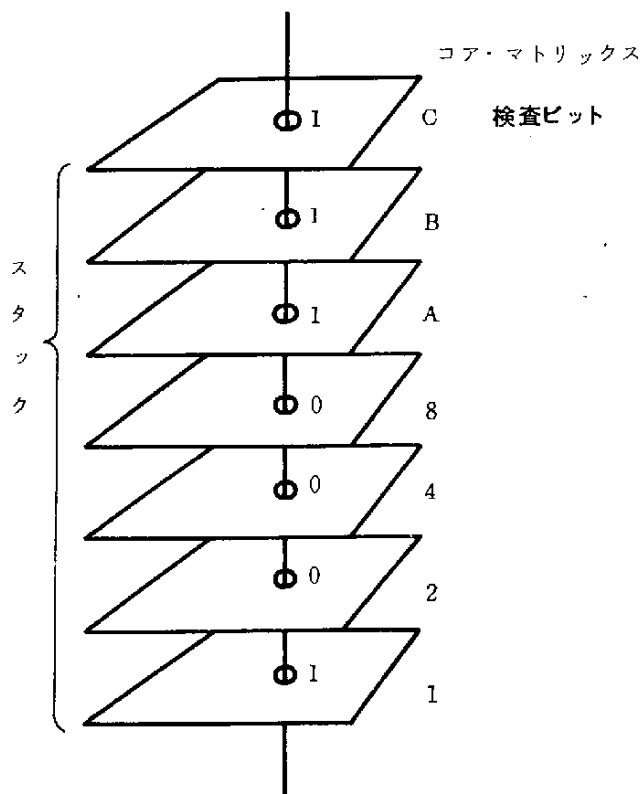
- (a) 記憶の原理



(b) コア・マトリックス



(c) スタック



主記憶のサイクル時間とアクセス時間（呼出し時間）について説明する。ナノ秒 マイクロ秒などの単位をあげて説明するとよい。

磁気コア記憶の特徴について経済性、アクセスの速さ、信頼性、融通性などの点を説明する。

命令の中でデータを指定する方法を語単位記憶と、字単位記憶に分けて説明し、字単位の記憶では次の2つの方法を紹介する。

(a) データの下位けた(単位けた)を指定する。

例 2 5 3 4 0
 ↑ ↑
 ワード・マーク 命令で指定する番地

ワード・マークまでのデータのけた数である。

(b) データの上位けたを指定する。

例 2 5 3 4 0
 ↑
 命令で指定する番地

命令の中では、番地の他にデータのけた数も指定する。

(2) 磁気フィルム記憶

磁気薄膜フィルム (magnetic thin-film), 磁気薄膜フィルム・ロッド・メモリ (magnetic thin-film rod memory) の原理を簡単に説明する。

磁気フィルム記憶は、磁気コア記憶に比較して小型であり、高速である。信頼性も高い。

3.3 補助記憶

主記憶との関連で補助記憶の役割を説明する。補助記憶の内容は、主記憶に読み込まなければ処理できない。補助記憶として、磁気コア記憶や磁気フィルム記憶を使うこともできるが、そのほかに次のような装置がある。

磁気ディスク記憶

磁気ドラム記憶

磁気カード記憶

磁気テープ記憶

磁気ディスク記憶、磁気ドラム記憶、磁気カード記憶を等速呼出し記憶と呼ぶこともある。

以上のそれぞれについて説明する。

(1) 磁気ディスク記憶

磁気ディスクの構造と機能を説明する。等速呼出し(直接アクセス)ができる装置であることを理解させることが重要である。次のそれぞれについて説明する。

(a) トラックとセクタ

(b) シリンダの概念

(c) アドレス

(d) ディスク・バック

(e) アクセス・アーム

(f) シーク時間と回転待ち時間

(g) データ転送時間

(h) アクセス時間

(i) 記憶容量

(2) 磁気ドラム記憶

磁気ディスクと対比させながら、磁気ドラムの仕組みを説明する。

初期のコンピュータでは、磁気ドラムが主記憶として使われていた。現在ではその高速性を生かして、ひんばんに使用されるデータを記憶しておくための補助記憶として用いられている。

記憶容量と速度について説明する。

速度については、次の点を述べる。

(a) 回転待ち時間

(b) データ転送時間

(c) アクセス時間

磁気ディスクと比較するとよい。

(3) 磁気カード記憶

磁気カードの仕組みとアクセスの仕方を説明する。

磁気カードは、他の補助記憶と比較して、次のような特徴を持っている。

(a) 記憶容量が大きい

(b) 低速である

(c) 価格が安い

以上の特徴を生かして、大容量のデータ・ファイルを必要とするオンライン・システムなどに使われていることを説明する。

(4) 磁気テープ記憶

磁気テープについては、第2章の入出力装置のところの説明したので、他の補助記憶と比較しながら、簡単に復習するとよい。

他の補助記憶は、等速呼出し（直接呼出し）ができるのに対して磁気テープ記憶では、順次処理（順次アクセス）しかできないことを説明する。

3.4 外部記憶

外部記憶は、入出力装置や補助記憶装置にセットすると、コンピュータで制御できるようになり、それらの装置から取りはずすとコンピュータで制御できなくなる。外部記憶として、次のようなものをあげ、入出力装置、補助記憶の復習をかねて簡単に説明する。

- (1) 紙カード、紙テープ、磁気テープ
- (2) 磁気インク文字用ドキュメント、光学文字用ドキュメント
- (3) 磁気カード、磁気ディスク・パック

指導上の留意点

データの表現については、2進数に親しませることが重要である。理解度を確かめるために、次のような問題を解いてみるとよい。

- (1) 次の2進数は10進数でいくつか。

- (a) 1010011
- (b) 11101
- (c) 110010

- (2) 次の10進数は2進数でいくつか。

- (a) 255
- (b) 83
- (c) 37

主記憶については、命令の中でデータをどう指定するかをよく理解させる必要がある。語単位の記憶と字単位の記憶の違いを、演算能力と多種多様のデータを扱える能力などの点から説明する。コンピュータの利用分野とも関連させて説明するとよい。

補助記憶については、それぞれの特徴を教え、どのように使い分けられているかを理解させることが重要である。磁気ディスク、磁気テープについては、簡単な計算の要素を与え、読み書きの時間を計算させてみてよい。

第4章 演算・制御装置

この章では次の用語を教える。

アドレス変更(修飾), アドレス・レジスタ, アドレス部, 移動(する), オペランド, 解読器(デコーダ), 基数, けた上げ, けた移動(シフト), 固定小数点, 指標(インデックス), 指標レジスタ(インデックス・レジスタ), 16進数, 正規化, 2倍精度, 比較(する), 浮動小数点, 編集(する), 補数, 命令, 命令レジスタ, 累算器(アキュムレータ), 累乗(べき), レジスタ, 論理命令, 記憶レジスタ, 汎用レジスタ, 浮動小数点レジスタ, 制御装置, 算術論理演算装置, 命令部, 命令サイクル, 実行サイクル, コマンド, 仮数, 符号の位置, 分岐

目 標

中央演算処理装置は、コンピュータの心臓部であり、コンピュータを構成している全装置を制御し監視する。主記憶の番地付け、情報の管理、情報の出し入れ、命令の解読と実行、主記憶と他の装置の間の情報のやりとり、データの算術演算、論理演算などを行なう。この章では、これらの機能を説明し、中央演算処理装置の役割を理解させる。

内 容

4.1 演算の原理

コンピュータの主記憶の中では、一般にデータは2進数の形式で表現されている。しかし、入出力の媒体では、10進数や文字の形でデータを扱わなくてはならない。そこで、2進数表現と10進数や文字表現との間の変換が必要になる。コンピュータは、この変換を自動的行なっている。次のような演算と数値の変換について説明する。

- (1) 2進数演算と数値の変換
- (2) 16進数演算と数値の変換

(1) 2進数演算

演算の仕組みを10進数と比較しながら説明する。1を越えるとけたが繰上がる(けた上げ)。

次の演算について説明する。

- (a) 加 算
- (b) 減 算
- (c) 乗 算
- (d) 除 算

減算に関連して、補数の説明をし、減算は減数の補数と被減数との加算によって行なうことを説明する。乗算は加算の繰返し、除算は補数による加算の繰返しであることを理解させる。

10進数から2進数、2進数から10進数への変換の仕方を説明する。

(2) 16進数演算

制御装置を有効に使えること、より多くの種類の文字を表現できることなどの理由で、新しいコンピュータでは、2進数に代って16進数表現が多く使われるようになった。次のような数の変換と演算について説明する。

- (a) 10進数から16進数への変換
- (b) 16進数から10進数への変換
- (c) 2進数と16進数の間の変換
- (d) 16進数演算

加算、減算、乗算、除算を説明し、簡単な練習問題を行なう。

4.2 中央演算処理装置の基本要素

次の3つの要素を説明する。

- ・レジスタ
- ・解読器(デコーダ)
- ・加算器

(1) レジスタの働き

レジスタの働きとして、データを受取る、データを保持する、データを移すという3つの働きを説明する。

次のような各種のレジスタについて説明する。

- (a) 累算器(アキュムレータ)
- (b) 記憶レジスタ
- (c) アドレス・レジスタ

- (d) 命令レジスタ
- (e) 指標レジスタ(インデックス・レジスタ)
- (f) 汎用レジスタ
- (g) 浮動小数点レジスタ

インデックス・レジスタに関連して、アドレス変更(修飾)の概念を説明する。インデックスの意味とインデックス・レジスタの役割とを理解させる。

(2) 解読器, 加算器

解読器と加算器については、簡単に説明する程度でよい。

4.3 制 御 装 置

制御装置の働きについて、次のような点を説明する。

- (1) 入出力装置の制御
- (2) 記憶装置からの情報の出し入れ
- (3) 記憶装置と算術論理演算装置の間の情報の出し入れ
- (4) 算術論理演算 操作の指示

以上のような制御装置の働きは、すべて主記憶に記憶されているプログラムによって制御されていることを説明する。入出力コマンドに触れる。プログラムについては、第3部コンピュータのソフトウェアのところで詳しく説明するので、ここでは次のようなことを述べればよい。

(1) プログラムの定義

プログラムは、処理手順を指示した一連の命令である。

(2) 命令(instruction)の構造

命令部とオペランド(アドレス部)から成っている。

それぞれの役割を簡単に説明する。

(3) 命令の実行の仕組み

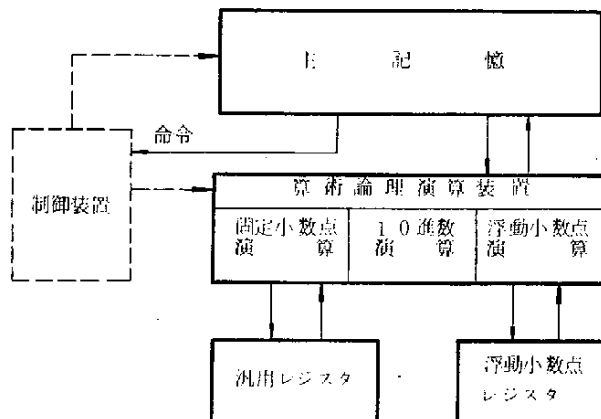
命令サイクル(Iサイクル)と実行サイクル(Eサイクル)に触れる。

4.4 算術論理演算装置

算術論理演算は、加算、減算、乗算、除算などの算術演算と、比較などの論理演算を実際に行なうところである。コンピュータは、多種多様なデータを扱う事務処理と、複雑な計算が要求される技術計算との、両方の要求に答えるために、次のような4種類の演算を行なっている。

整数演算	可変長	事務計算向き
固定小数点演算	固定長	技術計算向き
浮動小数点演算	固定長	技術計算向き
論理命令		

次のような図を書いて、主記憶と算術論理演算機構、レジスタ、制御装置の関連を説明する。



それぞれの演算について次のような点を説明する。

(1) 整数演算

可変長のパックされた数字を扱う。

storage-to-storage 方式の演算を行なう。

小数点と符号の位置、符号の取扱いについても説明する。

(2) 固定小数点演算

固定長の2進数を扱う。

storage-to-accumulator

accumulator-to-accumulator 方式の演算を行なう。

小数点と符号の位置、符号の扱い方について説明する。

(3) 浮動小数点演算

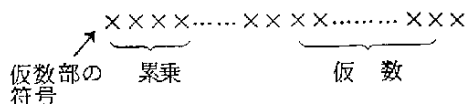
浮動小数点を使った数値の表現の仕方に親しませる。

次のような図を示すとよい。

$$0.42350 \times 10^5$$

仮数 基数、底 累乗
 小数部 (radix) 指数部
 (mantissa) (power)

浮動小数点演算は、浮動小数点レジスタで行なわれる。浮動小数点レジスタでは数値が次のように表現される。



正規化 (normalize) と 2倍精度について触れる。

固定小数点演算と比較して、浮動小数点演算の利点を述べる。

(4) 論理命令

AND, OR, Exclusive OR などは、理解が困難なので説明しない。

算術演算以外の演算として次のような機能を説明する。

(a) けた移動(シフト)

シフト機能の意義を説明する。

(b) 論理操作

Load, Store, Move(移動), Compare(比較), Branch(分岐)など。

(c) 編集機能

Pack, Unpack など。

(d) 2進10進変換 10進2進変換,

指導上の留意点

演算制御装置は、コンピュータの中で最も重要な構成要素の1つであるが、それだけに理解が困難な部分である。細かい点に立入りすぎると理解を困難にするだけであるから、基礎的なことにとどめ、概念を理解させるようにする。

ハードウェアの構造は、コンピュータによって異なるので特定のコンピュータを適当に選んで、そのコンピュータを一貫して取上げるようにするとよい。

演算は加算を中心に説明する。10進数と対比しながら説明するとよい。2進数の加減乗除演算も、人が10進数で演算するときと同じ方法で行なえるが、コンピュータでは加算を基本にして算術演算を行なうということから、独特の演算方法をとっていることを理解させる。

算術演算は、10進数演算と固定小数点演算を中心に説明すればよい。浮動小数点演算にはあまり立入る必要はない。

参考文献

1. MODERN DATA PROCESSING

ROBERT R. ARNOLD

HAROLD C. HILL

AYLMER V. NICHOLS

昭44. 3,580円

JOHN WILEY & SONS, INC.

このカリキュラムの基本文献である。

2. INTRODUCTION TO ELECTRONIC DIGITAL COMPUTERS

HARBERT MAISEL

DONALD L. WRIGHT

昭44. 3,980円

McGraw-Hill

ゆきとどいた参考書である。

3. COMPUTER ORGANIZATION AND PROGRAMMING

C. WILLIAM GEAR

昭44. 5,180円

McGraw-Hill

1章 Introduction

2章 Hardware organization

6章 Input, output, and backup storageが参考になる。

4. 上級情報処理技術研修ガイドブック

コンピュータ総合(基礎編)

通商産業省編

昭45.3 2,600円

(財) 日本情報処理開発センター

5. 電子計算機の原理と構造

桑原守二 著

昭45.9 1,000円

電気学会

原理と構成についての解説がまとまっている。

6. 図説 電子計算機システム

鈴木光彦 著

昭43.11 980円

日刊工業新聞社

1章 ハードウェアが参考になる。

第3部 コンピュータのソフトウェア

教育の目標

コンピュータのハードウェアが進歩するとともに、ハードウェアの機能を活用することに対してソフトウェアが次第に重要な役割を担うようになった。現在では、もはや特殊な場合を除いてソフトウェアの知識がなければ、コンピュータを使用することができない。そこで、ここではソフトウェアについての基礎的な知識を習得させ、コンピュータに対してソフトウェアの果している役割を十分理確させることを目標にする。

内容のあらまし

	時間数
プログラムのあらまし	
プログラム記憶方式	4
機械語	5
プログラムの作り方	8
プログラミング・システム	
プログラミング・システムの構成と用語	3
サブルーチンとマクロ命令	3
入出力制御システム	2
サービス・プログラム	3
プログラミング言語の種類	2
ソフトウェアの発展	2
言語のあらまし	
アセンブラ	2
FORTRAN	4
ALGOL	1
COBOL	5
PL/I	4
RPG	2

第1章 プログラムのあらまし

この章では次の用語を教える。

記憶ダンブ、機械語、記号プログラム、言語翻訳プログラム、原始プログラム、コンパイラ、アセンブラ、コンパイル、作業域、実行用コンピュータ、出力域、セグメント、流れ図、デバグ、文書化(ドキュメンテーション)、入力域、プログラミング、プログラミング言語、プログラム・テスト、プロセッサ(翻訳ルーチン)、翻訳(する)、翻訳用コンピュータ、命令コード、命令セット、目的プログラム、モジュール、サブルーチン、ローディング、データ・アドレス、命令アドレス、装置アドレス、アセンブル、コーディング、実行プログラム、記号言語、条件つき分岐、丸め、マクロ命令、ファイル、論理判断

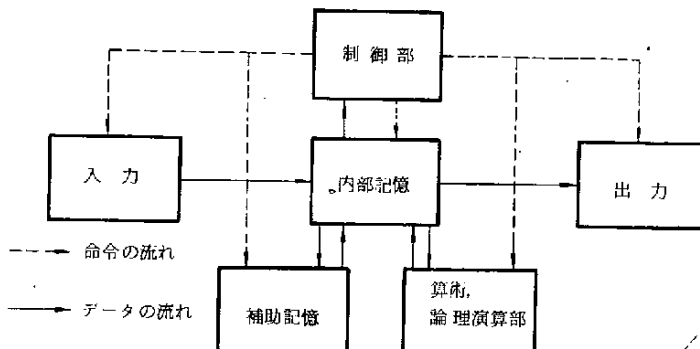
目 標

プログラム内蔵式のコンピュータで問題を解こうとすると、プログラミング—プログラムを作る—ということが必要になる。プログラムができて初めて、コンピュータで解を求めることができるのである。プログラムは、問題を解く処理手順を記述したものである。処理手順は、それぞれのコンピュータが持っている命令語を使って記述しなければならない。そこでこの章では、プログラムを記述する命令語について理解させ、プログラムがコンピュータに貯えられ実行される仕組み、プログラミングの仕方などを習得させる。

内 容

1.1 プログラム記憶方式

プログラム内蔵式コンピュータを構成している次のような機構について説明し、データ処理の



仕組みを理解させる。

プログラムについて、次のことを説明する。

- (a) プログラムの定義、プログラムと命令の違い
- (b) プログラム内蔵式の意味
- (c) 入出力媒体にあるプログラムと内部記憶にロードされているプログラムの区別
- (d) プログラムのローディングと実行
- (e) プログラムの実行順序

分岐 (branch) が起ると実行順序が変わることに触れる。

- (f) 実行の終り

実行を終りにする条件に触れる。

1.2 機械語 (machine language)

機械語について、次のようなことを説明する。

機械語の形式

機械語の種類

(1) 機械語の形式

最初に機械語の定義を説明する。機械語に対応して記号言語があることに触れておくとよい。

機械語は、コンピュータによって異なることを理解させる。機械語が命令部(命令コード)とオペランドから成ることを説明し、命令部とオペランドの意味を理解させる。加算、減算、乗算、除算、入出力開始命令等を例にあげて説明するとよい。

オペランドについては、オペランドが命令を実行するときに必要な

- (a) データの記憶されている場所
- (b) 次に実行する命令の位置
- (c) 使用する入出力機器の番号

などを指定するのに使われることを説明する。それぞれの位置は、アドレス(番地)を使って表わすこと、データ命令、入出力機器の場所を示すアドレスはそれぞれデータ・アドレス、命令アドレス、装置アドレスと呼ばれていることを説明し、それぞれの違いを理解させる。

特にデータ・アドレスと命令アドレスの違いをよく理解させることが重要である。

(2) 機械語の種類

命令セットについて簡単に触れ、命令を次の4種類に分類して説明する。

(a) 入出力命令 (input output instruction)

入出力命令について例をあげて説明する。編集 (editing) の機能にも簡単に触れる。

(b) 算術命令 (arithmetic instruction)

加算命令, 減算命令, 乗算命令, 除算命令を説明する。乗算と除算は, それぞれ加算と減算の繰返しで処理できることを理解させる。加減算の繰返しで乗除算を行なう処理手順に触れ, それが乗除算サブルーチンと呼ばれていることを説明する。サブルーチンについては第2章で詳しく説明することにして, ここでは簡単に説明すればよい。

演算に必要な要素と, 演算がレジスタや累算器などを使用して行なわれることを説明する。

けた移動と丸めの機能を教える。

(c) 分岐命令 (branch instruction)

分岐命令の意味を説明し, 無条件分岐 (unconditional branch) と条件付き分岐 (conditional branch) を理解させる。条件付き分岐は, 次の論理命令のところでも説明するので, ここでは言葉の意味程度でよい。

(d) 論理命令 (logic instruction)

論理命令の機能と, 重要性を説明する。論理命令の例として, 次のものを説明する。

データ転送 (data transfer) 命令

けた移動 (shift) 命令

比較 (compare) 命令

テスト (test) 命令

比較命令とテスト命令は条件付き分岐と関連づけて説明する。論理判断機能を理解させる。

1.3 プログラミングの仕方

プログラミングをする上での基本的な手順を6段階に分けて説明する。

プログラムの設計

プログラム流れ図の作成

プログラムのコーディング

アセンブルまたはコンパイル

プログラム・テストとデバグ

文書化

これらの6段階の作業について指導の内容をさらに詳しく述べることにしよう。

(1) プログラムの設計

プログラム作成の第1段階は、解くべき問題を明確にし、プログラムの機能や構成を決めることである。そのためには、問題のすべての局面を分析して見る必要がある。ここでは

- (a) 入力データの分析
- (b) 問題を解くのに必要な論理的、実的な処理の分析
- (c) 最終的な出力の形式の分析

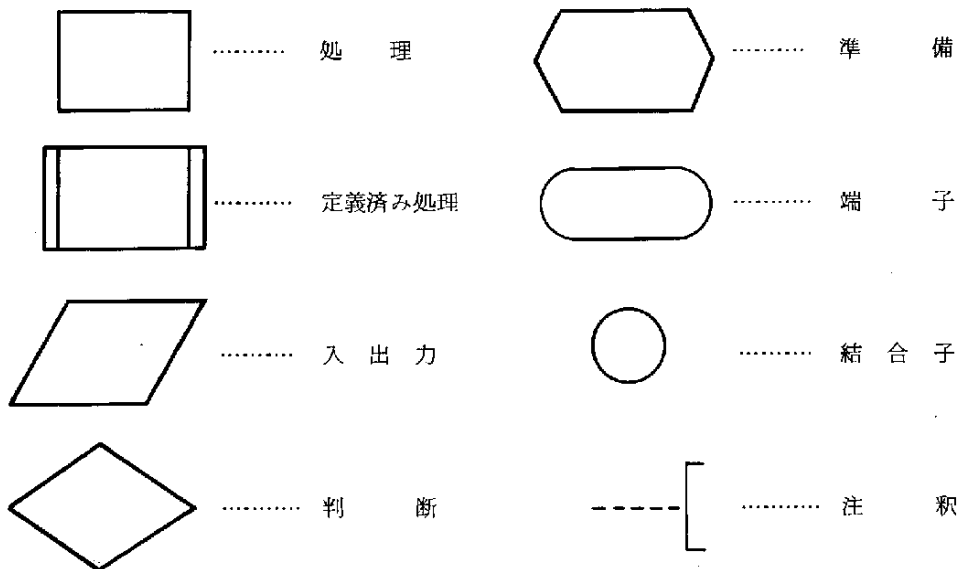
の3点について説明する。これらの分析の結果をもとにして、プログラム設計書が作られることにも触れる。

(2) プログラム流れ図の作成

プログラム流れ図の役割と書き方を理解させる。次のことを説明する。

- (a) 流れ図を書く目的と、流れ図を書くことによって得られる利点
- (b) 流れ図で使用する記号

流れ図記号はJIS規格によって標準的な書き方が決められていることを説明し（付録参照）、次の記号の使い方を理解させる。



(c) 上の記号を使用して、簡単な流れ図を例の示す。

(3) プログラムのコーディング

最初に次の語句と相互の関係を説明する。

- (a) プログラミング言語 (programming language)
- (b) 原始プログラム (source program)
- (c) 目的プログラム (object program)
- (d) 言語翻訳プログラム (language translator)
- (e) 実行プログラム (または処理プログラム) (processing program)
- (f) アセンブラ (assembler) とコンパイラ (compiler)

原始プログラムには、3つの型の命令が含まれていることを説明する。

- (a) 機械語に対応した 実行命令
- (b) マクロ命令
- (c) プロセッサへの指示命令

流れ図の説明のところで例示したものを適当な記号言語 (アセンブラ言語) でコーディングした記号プログラムを用意し、説明するとよいだろう。

次のことを簡単に説明する。

- (a) コーディングのとき使用する英数字の書き方
- (b) コーディング用紙の書式
 - (i) 名前欄
 - (ii) 命令欄
 - (iii) オペランド欄
 - (iv) 注釈欄
 - (v) 継続行
 - (vi) 識別欄
- (c) コーディング

コーディング例に従って、次のことを説明する。

- (i) プロセッサに指示を与える命令
- (ii) 使用するファイルを定義する命令
- (iii) 使用する入出力域や作業域と定数を定義する命令
- (iv) ファイルを開く (openする) 命令
- (v) 紙カードを読み内容を入力域に記憶させる命令
- (vi) 演算などの処理を指示する命令
- (vii) 出力域からの印刷を指示する命令

(viii) ファイルを閉じる (close する) 命令

(ix) プログラムの終りを示す命令

(4) アセンブルまたはコンパイル

原始プログラムをアセンブルしたときのリストを見せて、原始プログラムを目的プログラムに翻訳したときの出力について説明する。相互参照テーブルにも簡単に触れる。翻訳用コンピュータと実行用コンピュータの意味を説明する。

(5) プログラム・テストとデバグ

プログラム・テストについて、プログラム・テストの手順、プログラム・テストのときに考慮しなければならないことがらなどを説明する。

プログラム・テストのために必要なデータの準備の仕方を説明する。大きな複雑なプログラムは、セグメント(モジュール)単位のテスト、およびつなぎテストという2段階のテストを経ることを説明する。

エラー検出のため必要な記憶ダンプ(storage dump)の利用について説明する。プログラムの保守についても触れる。

(6) 文書化

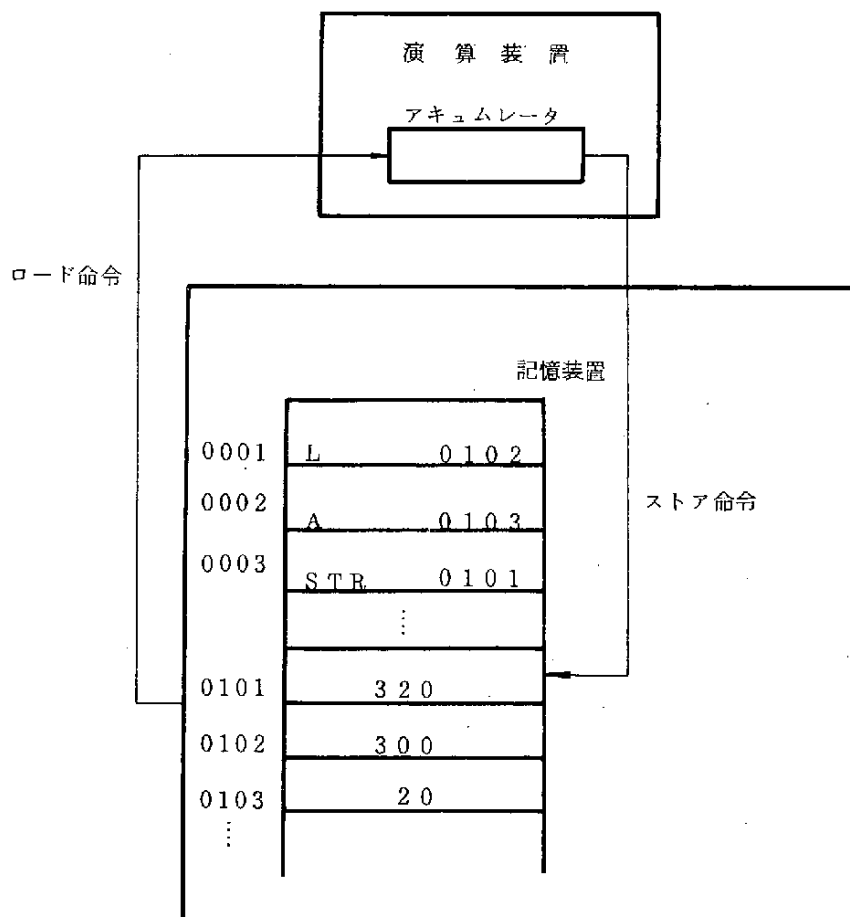
文書化の必要性を説明し、プログラム仕様書、プログラム設計書、操作手引書などには、どのようなものを載せたらよいかを説明する。情報処理用語や文書化に関連した標準規格について触れる(付録参照)。

指導上の留意点

(1) プログラム内蔵式概念を理解させるために、次のような方法が考えられる。

ロード命令をL、加算命令をA、ストア命令をSTRで表わすことにして、次のような図で命令の実行の仕方と、データの動きを説明する。

プログラムの説明では、プログラムが入出力媒体上にあるときは、データとして扱われること、内部記憶に記憶されているときにもデータとして扱われることがあることを説明して、プログラムとデータの区別について正確に理解させることが重要である。



(2) 機械語

機械語の説明で、命令部を説明するときは、次のような略語を使用するとよい。

ロード L

加 算 A

減 算 S

乗 算 M

除 算 D

入出力開始 S I O

ストア S T R

命令の説明をするときの1例を示そう。

L 5 1443 ①

A 5 2000 ②

S 5 1 5 2 0 ③

STR 5 1 4 4 3 ④

第1オペランドはレジスタの番号、第2オペランドは番地とする。①、②、③を実行したあと、1 4 4 3番地、2 0 0 0番地、1 5 2 0番地の内容がどうなるか、また④を実行したあとは1 4 4 3番地がどうなるかを説明する。

コンピュータにより、機械語の表現の仕方や形式はさまざまであること、また1ーアドレス方式、2ーアドレス方式、3ーアドレス方式、0ーアドレス方式などがあることも説明する。

(3) プログラムの作り方について説明するときには、プログラムを作成する手順の各段階における技法を説明するだけでなく、具体的な問題を取上げて、実際にプログラム作成の手順を踏んでみるとよい。例示する問題としては、次のようなものが考えられる。

- (a) 社員番号、名前、時間給、作業時間数をせん孔した紙カードを読み、給与を計算する。
- (b) 試験の採点結果を読み、合計点を計算する。
- (c) 各支店ごとの売上高を計算する。

これらの問題を処理していく段階を追いながら流れ図、せん孔された紙カード、アSEMBル・リストなどの実物を生徒に見せるとよい。

コーディング段階で、記号言語（アSEMBラ言語）を使うのは、コンピュータの内部構造や機械語のことを理解するのに問題向き言語よりも有効だと思われるからである。しかし、ここでは記号言語を教えることが目的ではないので、使用する命令は基礎的なものとどめる。

プログラム作成の段階では、プログラム・テストやデバグに多くの時間が必要である。これをできるだけ短時間で行なうために必要なことがらを説明し、テストの仕方やデバグについて理解させる。実例をいくつか紹介するのが望ましい。

第2章 プログラミング・システム

この章では次の用語を教える。

一括処理（バッチ処理）、入口点、応用プログラム、オペレーティング・システム、監視プログラム、擬似命令、降順、COBOL、サービスプログラム、ジェネレータ、主ルーチン、照合順序、昇順、出口、閉じたサブルーチン、パラメータ、PL/I、開いたサブルーチン、復帰点、FORTRAN、プログラミング・システム、ブロック化、文法検査（シンタックス・チェック）、まぜ合せ、モニタ、ユーティリティ・プログラム、呼出し（コール）、問題向き言語、インタプリタ（通訳ルーチン）、トランスレータ、ローダ、リンケジ（連結）、入出力制御システム、制御カード、準備ルーチン、データ変換プログラム、自動プログラミング、追跡ルーチン、記号アドレス、ニーモニック・コード、テープ・リール、ファイル・ラベル、ALGOL、RPG

目 標

現在機械語でプログラミングするということは、ほとんど行なわれていない。プログラミング用の言語が開発され、それらの言語を使ってプログラミングを行なうようになったからである。そこでこの章では、プログラミング言語、プロセッサ、サブルーチン、マクロ命令、サービス・プログラム、問題向き言語などの概念を説明し、プログラミング・システムについて基本的なことがらを教える。さらに、ソフトウェアという概念が生れた必然性を説明し、ソフトウェアの役割を理解させる。

内 容

2.1 プログラミング・システムの構成と用語

機械語でプログラムを書くと、人に大きな負担がかかる理由を説明し、人の負担を軽くし、機械への依存度を高める目的でプログラミング・システムが開発されてきたことを教える。

プログラミング・システムの中心は、プログラミング言語とプロセッサであるから、その2つについて説明する。

(1) プログラミング言語

プログラミング言語は、人が理解しやすい言葉でプログラミングできるようにという目的で開発されたもので、プログラミング言語の開発によってプログラミングが容易になったことを説明

する。プログラミング言語の進歩を歴史的にたどってみるとよいだろう。

(2) プロセッサ(翻訳ルーチン)

プロセッサの定義と役割を説明する。原始プログラム、目的プログラム、原始言語、目的言語について説明する。

プロセッサには、コンパイラ、アセンブラ、ジェネレータ、インタプリタ(通訳ルーチン)、トランスレータなどがあることを説明する。プロセッサが翻訳を行なうときの働きについて、次のような点を述べる必要がある。

- (a) 文法検査(シンタックス・チェック)とエラー・チェック
- (b) ニーモニック・コードを機械語に置き換える処理
- (c) オペランドに使っている名前(ラベル)と記憶装置の位置との対応づけ
- (d) 個々の命令に必要な記憶域の確保、記号アドレスと番地との対応づけ
- (e) 原始プログラムで要求される定数やデータの処理
- (f) 翻訳時に出力されるメッセージ
- (g) ロードの準備

プログラミング・システムの発達によって得られた利益を説明する。プログラミングが容易になったこと、デバグが容易になったことなどを例をあげて説明するとよい。自動プログラミングの概念にも触れる。

2.2 サブルーチンとマクロ命令

(1) サブルーチンの機能と特徴とを理解させるために次のことを説明する。

- (a) 主ルーチンとサブルーチンの関係、入口点、復帰点、出口などにも触れる。
- (b) 開いたサブルーチン(open subroutine)
- (c) 閉じたサブルーチン(closed subroutine)
- (d) 主ルーチンと閉じたサブルーチンとのリンケジ、サブルーチンの呼出し(call),
パラメータ受け渡し
- (e) サブルーチンの使用例

(2) マクロ命令(macro instruction)

マクロ命令の機能と特徴を理解させるために、次のことを説明する。

- (a) マクロ命令とは何か、マクロ命令を使うことによって、どのような利益があるか
- (b) 擬似命令(pseudo instruction)の説明、マクロ命令は擬似命令でできてお

り、開いたサブルーチンをそう入する働きをもっていること

(c) マクロ置換 (macro substitution)

(d) マクロ生成 (macro generation)

(e) マクロ命令の使用例

2.3 入出力制御システム (input output control system ; IOCS)

入出力制御システムの役割と、これを利用することによって得られる利益を説明する。

特に次のことを理解させる。

(1) 入出力制御システムが必要になった理由

(2) 入出力制御システムの一般的な働きとして

(a) 入出力時のエラー・チェックと回復

(b) テープ・リールの終了時 (EOR) とファイルの終了時 (EOF) に必要の処理手順の生成

(c) ファイル・ラベルの検査

(d) ブロック化と非ブロック化

(e) 入出力と演算とを同時処理するための制御

(3) 入出力装置とファイルの定義

(4) GET, PUT, OPEN, CLOSEなどのマクロ命令の使用

(5) 等速呼出し装置のためのIOCSの機能

2.4 サービス・プログラム

サービス・プログラムについて説明する。サービス・プログラムが開発された理由を述べ、次のようなサービス・プログラムについて説明する。

分類併合プログラム

その他の各種ユーティリティ・プログラム

応用プログラム

(1) 分類併合プログラム (sort and merge program)

ある問題を解決する過程で、データを並べ替えたり、**まぜ合せ**たりすることが必要になる場合がよくある。これは、数の大きい順 (昇順) のこともあれば、小さい順 (降順) のこともある。また、アルファベット順というようなこともある。このように処理に必要な順序 (照合順序) に並べる操作を行ってくれるのが分類併合プログラムであり、サービス・プログラムの1つとして最も早く開発され、利用されていることを説明する。特に事務処理の分野における分類併合プ

プログラムの役割について、事務処理の流れ図を示して、その使用頻度や、コンピュータの使用時間の中で占める割合などを説明し、その重要性を理解させる。

分類併合プログラムを使用するときには、最初に、一般的に作られている分類併合プログラムを最適化するという処理が必要なこと、最適化に必要な指示を、制御カードで与えることを説明する。制御カードの内容について、簡単な例を示すとよいであろう。

一般にサービス・プログラムを使うときには、制御カードによって情報を与え、処理に先だってプログラムを最適化したり、プログラムに処理の仕方についての指示を与えることを説明する。分類プログラムの構成について触れ、前処理のフェーズ、分類のフェーズ、最終的なませ合せのフェーズという3つのフェーズがあることを説明する。

汎用的な分類併合プログラムには、データを追加したり、変更する機能や、利用者のプログラムを付け加えて処理の仕方をかえる機能などもあることに触れ、それがサービス・プログラムが一般的にもっている機能であることを説明する。

(2) その他の各種ユーティリティ・プログラム (utility program)

分類併合プログラムのほかに、次のような各種のユーティリティ・プログラムがあることを述べ、それぞれの役割と使用例を説明する。

(a) 準備ルーチン (housekeeping routine)

(b) データ変換プログラム (入出力変換プログラム) (data conversion program)

(例) 磁気テープから磁気ディスク (tape to disk) への変換、紙カードから磁気テープ (card to tape) への変換、磁気テープ複写

(c) プログラム・テストの補助プログラム (追跡ルーチン)

(d) 技術計算用ライブラリ

(3) 応用プログラム (application program)

特定の適用分野や業務と結びついた汎用的なプログラムとして、応用プログラムの開発が進んでいることを説明する。在庫管理や工程管理のためのプログラム、評価と予測のためのプログラムなど2,3の例をあげて、その役割を説明する。

2.5 プログラミング言語の種類

プログラミング言語には、機械向き言語 (machine oriented language) と問題向き言語 (problem oriented language) がある。よく使われている問題向き言語を取上げて、両者を対比させ、それぞれの特徴を説明する。個々の言語については、第3章で説明

するので、ここでは、FORTRAN, ALGOL, COBOL, PL / I RPGなどのことばに親しませる程度でよい。

対比する項目の例

- (a) 文法上の特徴
- (b) 翻訳の出力
- (c) 翻訳速度、実行速度
- (d) 機械語との対比
- (e) 習得のしやすさ
- (f) プログラムの修正やデバグの難易
- (g) 用 途

2.6 ソフトウェアの発展

プログラミング・システムのまとめをし、プログラミング・システムとの関連でソフトウェアの概念と役割を説明する。

ソフトウェアについて理解させるために、次のような点について説明するとよい。

- (a) 情報処理の拡大
- (b) 情報処理に対するコンピュータの役割
- (c) プログラミング・システムの発展
- (d) システム分析やシステム設計、システム評価、文書化などの技法の開発
- (e) プログラミング技法の開発
- (f) コンピュータ使用経験の蓄積
- (g) コンピュータの運営、管理についての経験の蓄積
- (h) 情報処理技術教育の進歩発展

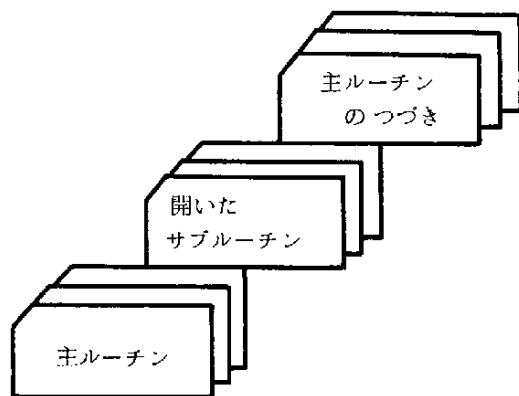
ソフトウェアの重要性を理解させるために、公表されている数字を使って、コンピュータを開発する際のハードウェアのコストとソフトウェアのコストを対比させて説明してもよい。

時間があれば、プログラミング・システムの発展した形態としてオペレーティング・システムの概念を説明する。またコンピュータの適用分野の広がりについて、近い将来の予想を含めて簡単に説明する。

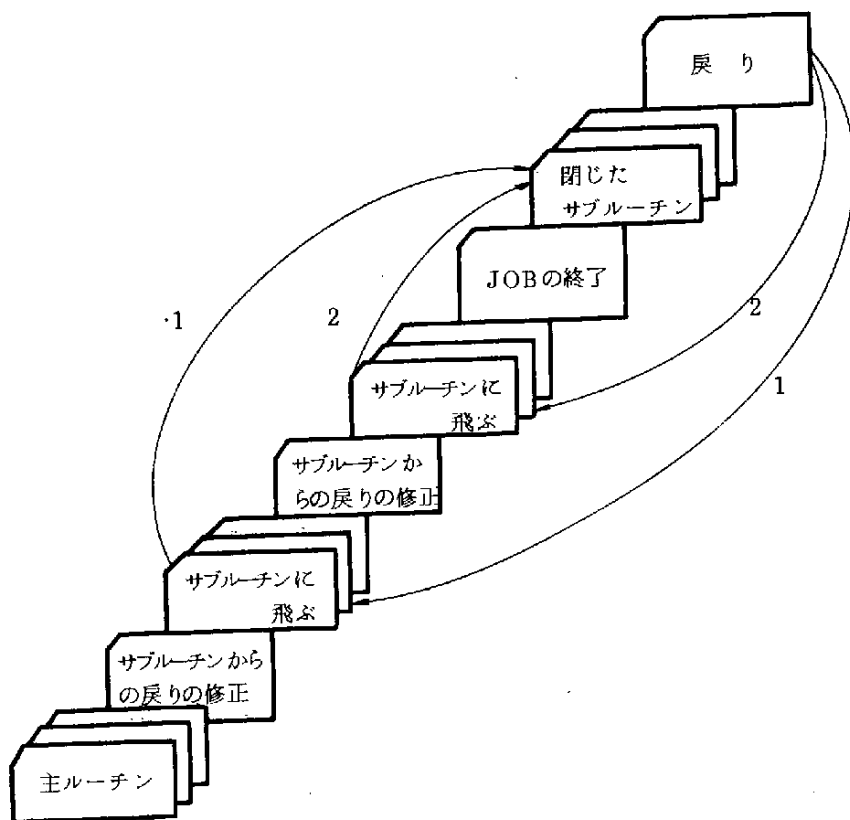
指導上の留意点

開いたサブルーチンと閉じたサブルーチンの説明では次のような図を用いるとよい。

開いたサブルーチン：



閉じたサブルーチン：



マクロ命令の例では、第1章で使ったアセンブル・リストを示して、入出力のマクロ命令とそれが展開されて生成された部分とを対比して説明するとよい。I O C Sについても第1章で使ったアセンブル・リストを示して、ファイル定義のマクロ命令と、それが展開された例を説明する。このように実際に目を通して理解を深めることが必要である。

分類プログラムの使用例としては、企業内の月末の経理処理や報告書作成処理の例をあげて、同一のファイルから分類の仕方だけを変えて、数多くの表を出力する必要性を述べるとよい。

併合プログラムの使用例としては、週ごとに整理されている磁気テープ・ファイルを月単位でまぜ合せたり、月ごとのファイルを期ごとにまぜ合せたりする例をあげるとよい。

分類プログラムが行なう分類の仕組みは、プログラミングの技法としても意味が深いので、分類の処理の仕方をたどってみて、プログラムの論理について生徒に興味をいだかせ、関心を深めるのもよい。時間があれば、次のような例をあげて、処理をたどってみよう。

磁気テープAに入っている次のような数字を昇順に並べ替える。

5 1 9

6 7 6

0 1 7

7 7 3

3 7 2

3 4 8

3.0 1

5 2 6

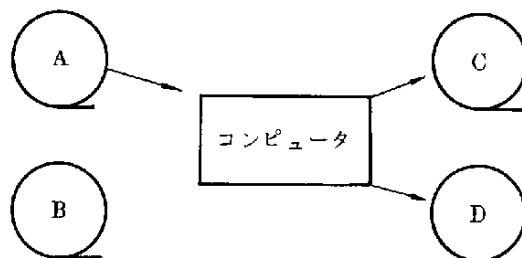
4 1 6,

8 8 8

2 5 5

1 0 1

フェーズ1では3個の数字ごとに処理する。



テープC

0 1 7 }
5 1 9 } c
6 7 6 }

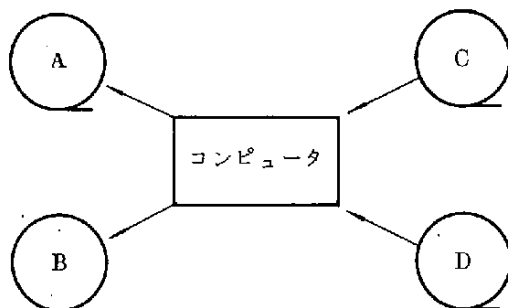
テープD

3 4 8 }
3 7 2 } d
7 7 3 }

3 0 1 }
4 1 6 } c'
5 2 6 }

1 0 1 }
2 5 5 } d'
8 8 8 }

フェーズ2ではまぜ合せ 処理を繰返す。cとd, c'とd'をまぜ合せ て、それぞれ磁気テープA, Bに書く。cとc'の境、またdとd'の境は数字が逆順になったところである。このような場所が完全になくなり、2個の磁気テープのそれぞれの数字が全部昇順に並ぶまで、磁気テープを巻戻しては処理を繰返す。



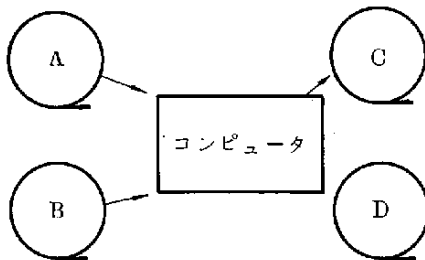
テープA

0 1 7
3 4 8
3 7 2
5 1 9
6 7 6
7 7 3

テープB

1 0 1
2 5 5
3 0 1
4 1 6
5 2 6
8 8 8

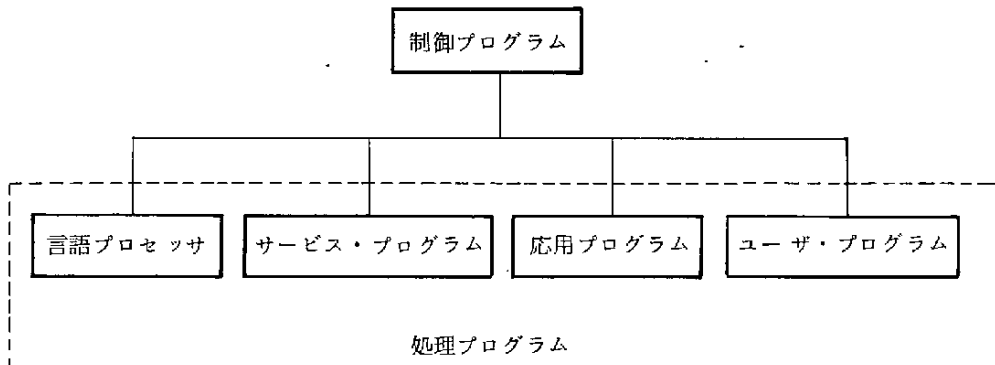
フェーズ3では、最後のまぜ合せ 処理をして、1つの磁気テープに書く。



テープC（最終結果）

0 1 7
 1 0 1
 2 5 5
 3 0 1
 3 4 8
 3 7 2
 4 1 6
 5 1 9
 5 2 6
 6 7 6
 7 7 3
 8 8 8

オペレーティング・システムの説明をする場合には、オペレーティング・システム開発の歴史に触れ、次のような図を示して説明する。



制御プログラムについては、正確に理解させる必要はない。初歩的なモニタ（監視プログラム）の登場から始め、発展の過程を説明して、一括処理（バッチ処理）を中心にして、制御プログラムの役割を理解させればよい。

サービス・プログラムの代表的なものの1つに、オペレーティング・システムとの関係で重要なリンケジ・エディタがある。リンケジ・エディタについて正確に理解させることは困難だと

思われるので、説明する場合には説明の仕方を、特に工夫する必要がある。

簡単な機能を持つ小さなプログラムを組合わせて、複雑な機能を持つ大きなプログラムを作ることができる技法などを簡単に説明し、プログラムの結合や編集という処理を理解させた上で、リンケジ・エディタの役割を説明したらよいだろう。

第3章 言語のあらまし

目 標

この章の目標は、第2章で触れた機械向き言語と問題向き言語について、広く使われているものを取上げて、言語の特徴と簡単な記述形式を習得させることである。

内 容

3.1 アセンブラ

第1章のプログラムのコーディング例を参照しながら、一通り簡単なアセンブラ言語を説明する。問題向き言語を説明するときは、アセンブラ言語と対比してみるとよい。

3.2 FORTRAN

FORTRANの歴史と、FORTRANの特徴を説明する。次のような点に重点をおいて教えることにする。

- (1) FORTRANの歴史、FORTRANの語源
- (2) 技術計算向きの言語であること

FORTRANの文について例を見せながら、簡単に説明する。

- (a) 入出力文

READ文、WRITE文を説明する。

- (b) 制御文

算術IF文、無条件GO TO文を説明する。

- (c) 算術代入文

算術式と似ていることや、算術式との違いなどを述べる。

- (d) FORMAT文

READ文、WRITE文と関連させて、FORMAT文を説明する。

- (e) 手続きと副プログラム

関数とサブルーチンを説明する。

FORTRANのコンパイルの仕組みについて、簡単に説明する。FORTRAN言語からアセンブラ言語にいったん変換するものと、FORTRAN言語から直接機械語に変換するものと

があることに触れるとよい。

3.3 ALGOL

ALGOLとFORTRANとを比較しながら、その特徴を説明する。

- (1) ALGOLの歴史、語源を説明し、ALGOLの先駆をなしたIAL (International Algebraic Language) に触れる。
- (2) FORTRANはアメリカで発達したが、ALGOLはヨーロッパで発達したことを説明し、利用状況等を述べる。
- (3) FORTRANは技術計算向きの言語であるのに対して、ALGOLはより純粋数字向き言語であることを述べる。
- (4) ALGOLとFORTRANを対比させて、文法上の違いをおおまかに説明する。

3.4 COBOL

COBOLの特徴を理解させるために、次のことがらを説明する。

- (1) COBOLの歴史と語源
- (2) 事務計算向きの言語であること
- (3) COBOLの言語体系
- (4) COBOLプログラム・シートへの記入の仕方
- (5) 4つのDIVISIONの説明

例を示して、例に従いながら説明する。

- (a) 見出しディビジョン

(IDENTIFICATION DIVISION)

- (b) 環境ディビジョン

(ENVIRONMENT DIVISION)

- (i) 構成セクション

(CONFIGURATION SECTION)

- (ii) 入出力セクション

(INPUT-OUTPUT SECTION)

- (c) データ・ディビジョン

(DATA DIVISION)

- (i) ファイル・セクション
(FILE SECTION)
- (ii) 作業域セクション
(WORKING-STORAGE SECTION)
- (d) 手続きディビジョン
(PROCEDURE DIVISION)
- (6) 命令 (statement), 文 (sentence), 段落 (paragraph), セクション (section) の関係
- (7) COBOLの利点と問題になる点
- (8) 次のような命令の種類と意味
 - (a) 翻訳指示命令
 - (b) 無条件実行命令
 - (c) 条件命令
 - (d) 注釈命令

3.5 PL / I

PL / I の特徴や、プログラムの簡単な記述の仕方を理解させるために、次のことに注意して説明する。

- (1) PL / I 言語の開発された背景と語源, COBOL, FORTRAN, ALGOL などの関係
- (2) プログラムの記述の形式に COBOL や FORTRAN のような厳しい規則がないこと
- (3) ブロック構造と PROCEDURE ブロック
ブロックの持つ利点
- (4) PL / I プログラムは、語と区切符号から成ること
 - (a) 語では名前と定数の説明をする。
 - (b) 区切り符号では、コンマ、セミコロン、ピリオドなどのセパレータ (separator) と、演算子について触れる。
演算子では次のものを説明する。
 - (i) 算術演算子
 - (ii) 比較演算子

④ 論理演算子

(5) 注釈の書き方

プログラム例を示して、次の説明をする。

(1) DECLARE文

名前に属性を与える方法について説明する。

PL/Iで扱う変数、データの構造、配列などについて簡単に説明する。

COBOLのPICTURE指定と比較する。

(2) ファイルのOPEN

(3) READ文, WRITE文

(4) データの移動, 演算文

(5) GO TO文

(6) ENDFILE条件

(7) FILEのCLOSE

さらにCOBOLやFORTRANと比較しながら、類似点や相違点、利用分野や利用度、開発の方向などを説明するとよい。

3.6 RPG

RPGの特徴を、次のことに重点をおいて説明する。

(1) ジェネレータの概念

アセンブラ、コンパイラと比較して説明する。特に人が与えた条件に従って、コンピュータがプログラムを生成するという考え方を理解させることが重要である。

(2) RPGの開発された理由と開発の歴史

(3) RPGの持っている次の2つの機能の説明

(a) プログラムの生成

(b) データ処理

(4) RPGを使うときの作業手順

(5) RPGの4つの仕様書の役割と内容の説明

例をあげて簡単に説明すればよい。

(a) ファイル仕様書 (file specification sheet)

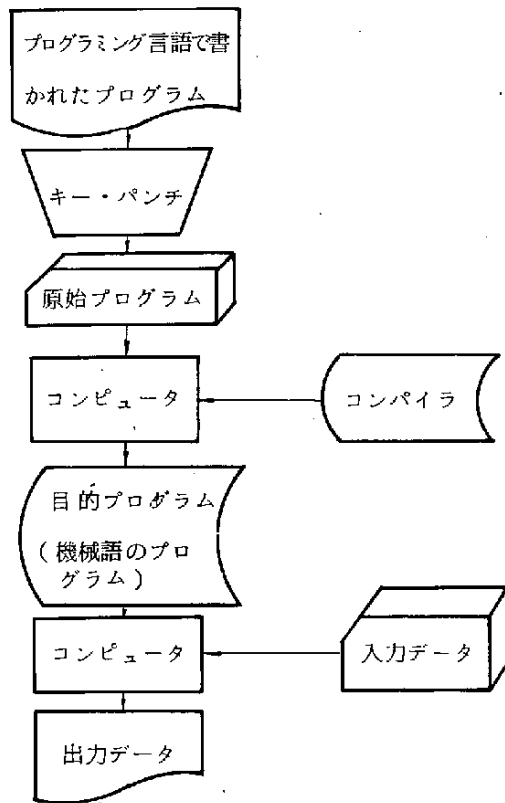
(b) 入力仕様書 (input specification sheet)

- (c) 計算仕様書 (calculation specification sheet)
- (d) 出力仕様書 (output-format specification sheet)

指導上の留意点

プログラム言語を使ってプログラムを書き、実行を行なうまでの手順を流れ図を使ってよく理解させる。

流れ図の例



コンパイルの仕方として、問題向き言語 → 記号言語 (アセンブラ言語など) → 機械語のように2段階で変換する方法と、原始言語から直接機械語に変換する方法とがあるが、前者の方が一般的である。

この章では、多くの言語を説明しなければならないが、第1章のプログラムのコーディングの項で使用したプログラム例と同じ問題を取上げて、それぞれの言語でコーディングをして比較するのがよい方法である。同じ問題を、多くの異った言語でコーディングをしてみせることによ

て、その言語がもつ特徴をよく理解させることができるであろう。

この章は、生徒がそれぞれのプログラミング言語を使ってコーディングできるようにすることを目的としているのではないから、プログラムが書けるようになる必要はない。おもなプログラミング言語の特徴を理解させ、それがどのようなときに使われるかが理解できればよいのである。

コーディングの実例として、コーディング用紙に記入した例やコンピュータで問題を解いた例などを生徒に見せるとよい。コンピュータで処理した例としては、正しい結果ばかりでなく、エラー・メッセージの出ているものなども見せた方がよい。

ソフトウェアについての説明が終ったところで、おもなプログラミング言語を比較検討させて、特徴や利用分野、言語のあらましなどをまとめさせ、生徒の理解度を確かめてみるのが望ましい。

参考文献

1. MODERN DATA PROCESSING

ROBERT R. ARNOLD

HAROLD C. HILL

AYLMER V. NICHOLS

JOHN WILEY & SONS, INC.,

昭和44 3,580円

このカリキュラムの基本文献である。

2. INTRODUCTION TO ELECTRONIC DIGITAL COMPUTERS

HARBERT MAISEL

DONALD L. WRIGHT

McGraw-Hill

昭和44 3,980円

PART 2 FUNDAMENTALS

PART 4 THE FORTRAN IV AND PL/I

PROGRAMMING LANGUAGESが参考になる。

3. COMPUTER ORGANIZATION AND PROGRAMMING

C. WILLIAM GEAR

McGraw-Hill

昭44 5,180円

第3章 System software. 第4章 Principles of
programming

4. 改訂コンピュータ概論

梶沢 昌和著

ダイヤモンド社

昭45.2 1,500円

V ソフトウェア, VI プログラミング, VII 各手法によるプログラミング

5. 電子計算機最新プログラミング

竹下 亨著

日本経営出版会

昭43.5 2,400円

第I部 電子計算機の基本知識

第II部 プログラミングの基本

第III部 問題向きプログラミング言語

第IV部 機械語から日常語への接近

第V部 ドキュメンテーションやプログラム・テストの手法

6. 情報処理受験講座 ソフトウェアの知識

矢田 光治著

オーム社

昭45.7 850円

ソフトウェア全般にわたっての基礎的な概念に触れている。

7. 電子計算機論

小南 尚義著

共立出版株式会社

昭43.1 700円

第6章 電子計算機のソフトウェアが参考になる。

8. コンピュータ処理の基礎知識

日本ソフトウェア(株)編

電機大学出版局

昭4 5.7 380円

第6章から第12章までにソフトウェアの基礎を載せている。

9. 電子計算機ソフトウェアの基礎

ハーバード・D・リード

ジェラルド・M・ワインバーグ共著

水野 幸男訳

培風館

昭4 1.1 1 1.800円

COMPUTER PROGRAMMING FUNDAMENTALSを翻

訳したもので、初心者向きにソフトウェアの基礎を説明している。

第4部 コンピュータの利用

教育の目標

- (1) システム設計の必要性和、どのような手順でシステム設計を行なったらよいかを理解させる。
- (2) システム設計とプログラム設計の関係、特にシステム流れ図とプログラム流れ図との関連と、それぞれの役割を理解させる。
- (3) 機械化についての指示書（仕様書）を見て、システム流れ図、コード設計書、ファイル設計書を作成できる能力を養う。
- (4) 情報処理規格について理解させる。
- (5) コンピュータがどのような分野で、どのような業務に利用されているかを説明し、コンピュータの応用について理解を深める。

内容のあらまし

コンピュータを利用するために必要なシステム設計の方法と、コンピュータがどのように利用されているかというコンピュータの応用について、次のように分類して説明する。

内 容	講義（時間）	演習（時間）
システム設計		
システム設計の手順	7	4
コード設計	3	
ファイル設計	6	
流れ図	5	
プログラム設計	5	7
情報処理規格	3	
コンピュータの応用	10	
合 計	39	11

第1章 システム設計

この章では次の用語を教える。

カード・イメージ、合計印刷(グループ印刷)、検査数字、コード、固定長レコード、コーナ・カット、資源、システム、システム分析、ゼロ抑制、取引ファイル、ハッシュ合計、パディング、プログラマ、ブロック化因数、ボリューム、明細印刷、レコード、レコード長、論理レコード、システム設計、適用業務(アプリケーション)、システム流れ図、提案書、コード・ブック、分類項目、参照項目、計算項目、基本カード、取引カード、ボリューム・ラベル、可変長レコード、パディング・レコード、テンプレート、判断表、プロシージャ、ルーチン、プログラム流れ図、粗い流れ図、詳細流れ図、ISO、JIS、互換性、基本ファイル。

目 標

- (1) システム設計からプログラム設計にいたる手順を説明し、システム設計の意味を理解させる。
- (2) 流れ図記号を説明し、簡単なプログラム流れ図が書けるように指導する。
- (3) コードの種類と特徴を説明し、コード設計についての知識を持たせる。
- (4) カード・ファイル、磁気テープ・ファイル、印刷様式などの設計について理解させる。
- (5) 情報処理関係の標準規格にどんなものがあるかを理解させる。
- (6) 文書化の意義について理解させる。

内 容

1.1 システム設計の手順

最初に、第1部第1章「事務とデータ処理」で説明した内容を復習して導入とする。

システム設計では、事務処理システムの理解が基本になる。現在の事務処理の仕組みがどうなっているかを掌握し、コンピュータで処理するとそれがどう変わるかを検討することが、システム設計の第1歩である。

次のような点について説明する。

- (a) 事務処理システムの理解

(b) システム分析とシステム設計の目的

(c) システム設計の手順

それぞれの項目で説明すべき内容を次にあげる。

(1) 事務処理システムの理解

これには、次のような処理の仕方がある。

(a) 手作業による処理

(b) 加算器などの簡単な機械を使った処理

計算、集計、記帳が1度に行なわれる。

(c) パンチ・カード・システム (PCS) による処理

事務処理はかなり機械化される。

(d) 各種の機械を有効に活用して機械化した処理

パンチ・カード・システムのほか、タイプライタ、会計機、宛名印刷機などが使われる。

(e) コンピュータによる処理

パンチ・カード・システムで事務処理を行なうようになったときから、機械化の対象となる業務の事務分析が必要になった。コンピュータによる処理が進み、事務分析の対象が大きくなってシステム分析という考え方が生れた。

(2) システム分析とシステム設計の目的

最初にシステムの概念をできるだけ現実に即して、具体的に説明する。

システムに対応する日本語は組織である。組織ということばからは、システムの概念をつかみにくい。システムを、ある仕事を処理するために組織で、体系づけられた全体という程度の説明にして、システムを構成する要素を説明しながら、システムの概念を理解させるのが望ましい。

システムの要素として次のようなものをあげる。

(a) 対象となる事務

(b) 事務処理に必要な道具や設備

(c) 事務処理の手順、処理にたずさわる人

コンピュータが中心となるシステムでは、次のような要素が重要である。

(a) 適用業務 (アプリケーション)

(b) コンピュータ (ハードウェア)

(c) プログラム、入出力ファイル (ソフトウェア)

これらの要素をどうしたら最も効果的な方法で結びつけることができるかという検討が、シス

テムについての検討であるということが出来る。

システムを検討するために必要なシステムの分析とシステムの設計について、次のことを説明する。

(a) システム分析の目的は、新システムに含まれなければならない要件を正確に掌握することである。システム分析では、システム設計の各条件（仕様書や設計方針など）を検討し、何を入力し、どのような処理をし、何を出力するかというシステム設計に必要な3つの要素を掌握する。

(b) システム設計の目的は、基本方針に従ってコンピュータによる処理が中心の新しいシステムを作ることである。システムの設計にあたっては、ハードウェア、ソフトウェア、適用業務をどのように結びつけたらよいかという検討が重要である。

(3) システム設計の手順

次のような新システム開発の手順を説明し、システム設計の段階ではどのような作業が必要かを明らかにする。システムズ・エンジニア、プログラマ、コーダと呼ばれる人たちの作業分担について簡単に触れる。

- (a) システムの分析
- (b) システムの設計
- (c) プログラムの設計
- (d) プログラムの作成
- (e) 新システムの実施
- (f) システムの評価、改善

システムの設計で行なう作業としては、次のものをあげればよい。

- (a) システム流れ図の作成
- (b) システムで必要な資源を決める。

資源としては、使用する取引ファイル（入力ファイル）、出力ファイル、基本ファイルが特に重要である。

- (c) コードの設計
- (d) 入出力ファイル、基本ファイルの設計

システムの設計を広く考えると、新しいシステムの目標を決め、新システムの対象範囲を決めるという作業が含まれる。そのためには、まず現行システムについての資料を集め、現行システムを十分理解することが必要になる。現行システムの理解をもとに、新システムに盛り込む要件を

決め、設計方針をかためるということになる。

そうして設計方針がかたまると、それを提案書にまとめる。

しかし、ここではそのような広く考えたシステム設計の概念には簡単に触れる程度にして、あまり立入らないことにする。

プログラム作成に必要なことがらを決めるという限定された意味でのシステム設計ということにして説明することが望ましい。

システム流れ図については、例をあげて説明するとよい。

システム流れ図について、次のような点を述べる。

(a) システム流れ図では、作業全体の概要と個々の処理の関連を明らかにする。

(b) 個々の処理について、使用する入出力ファイル、基本ファイル、プログラムを記述する。

コードの設計、入出力ファイル、基本ファイルの設計については、次の節で説明する。

1.2 コード設計

コンピュータ利用の効率を高めるためには、情報の伝達、処理、分類、記録、検索などのそれぞれについて効率を高めることが必要である。そのために、コードは重要な役割を果たしている。

コード設計について、次の点を説明する。

- ・コードの意味
- ・コード作成の手順
- ・コードの種類と用法
- ・コード化の作業
- ・コードの誤り検査
- ・コード体系化の実例

それぞれの項目に盛込むべき内容を次に述べる。

(1) コードの意味

コードの定義、コード化の役割などを説明する。

コードとは、データ項目を使用目的に応じて識別したり、分類したり、配列したりするために使用する数字や文字、記号などである。

コード化の意義を説明する。だれでも理解できる、どこでも通用する、いつでも利用できるなどの条件を説明する。コード化の目的として

- (a) データの体系化
- (b) データ処理システムの効率化

(c) データの互換性のための標準化

などをあげる。

(2) コード作成の手順

コード作成の手順は、データを分析し、目的に応じたコードを設計し、コード表・解説などを記載したコード・ブックを作成するまでの手順である。

次の点を述べる。

(a) 処理すべきデータの体系化

(b) 適用するコードの種類の決定

(c) コード・ブックの作成

(d) コードの改訂増補

(e) コードの管理

(3) コードの種類と用法

次のような種類のコードを説明する。

(a) 順番コード(sequence code)

(b) 区分コード(block code)

(c) けた別コード(group classification code)

(d) 10進コード(decimal code)

(e) 表意コード(mnemonic name)

(f) 数字式文字コード(numerical-alphabetical code)

(g) 合成コード(combined code)

(h) 子音コード(consonant code)

(i) 発音コード(phonetic code)

(4) コード化の作業

コード化の作業の良し悪しが、データ処理システムの効率に非常に影響することを説明する。

次のようなコード化の作業の方法を説明する。

(a) 人手によるコード化作業

(b) コンピュータによるコード化作業

(c) コンピュータ以外の機器によるコード化作業

(5) コードの誤り検査

データ処理のどの段階でコードの誤りが発生するかを説明し、その対策を検討する。

次の点を述べる。

- (a) 誤り発生の原因
- (b) 誤り検査の方法

検査数字を使う方法，コードのハッシュ合計を使う方法，コードの対応検査，相関検査などについて述べるとよい。

- (6) コード体系化の実例

コード体系の具体例を紹介して，コード化のいきさつ，必要性，コード体系の構成，コード設定の仕方，コードの使用例などを説明する。

1.3 ファイル設計

ファイルの設計の仕方を説明する。設計の手順と設計にあたって必要な技法について理解させる。

説明が一通り終わったところで簡単な演習問題を与えて，理解度を確かめてみるとよい。

設計する対象としては，報告書（出力ファイル），紙カード（入力ファイル），原書類を取りあげる。磁気テープのファイル設計についても簡単に説明する。磁気ディスクは取扱わない。

データ処理の流れは，原書類→紙カード→報告書という手順になるが，設計の手順としては，出力する内容にどのようなものを盛込むべきか，それをどういう形式で出力するかという点をまず決めて，そのような出力を得るために必要な入力は何かということから紙カードの内容を決め，原書類を決める方が効果的である。

したがって，報告書の設計を最初に説明することにする。

- (1) 報告書設計

報告書設計用紙（spacing chart，付録参照）の記入の仕方を説明する。次のような点に触れる必要がある。

- (a) 明細印刷と合計印刷（グループ印刷）
- (b) 見出し印刷行，明細印刷行，合計印刷行，第1印刷行，最終印刷行
- (c) スペースとスキップの機能
- (d) データ項目の位置

分類項目，参照項目，計算項目などの位置

- (e) 編集の機能

ゼロ抑制

小数点とコンマのそう入

合計記号, 正負の符号

円記号

浮動円記号

(f) 印刷用紙

汎用的な一般用紙と、題字、見出し、わくなどが印刷されている特定用紙とがある。

(g) 報告書の様式とキャリッジ・テープの関係

生徒の理解を確かめるために、特定の業務の例を取上げ、印刷の仕方についていくつかの条件を与えて、生徒に報告書を設計させてみるとよい。

(2) カード設計

カード設計では、取引カード (transaction card) を例にとって説明する。

カード設計の手順について、次のように順を追って説明する。

(a) 報告書の項目を抜出す。

(b) 抜出した項目をコンピュータの中で作り出すことのできるもの (計算結果または定数として) と、入力データとして与えなければならないものとに分ける。

(c) 入力データとして与えなければならないものについて、基本カードにあるものと取引カードからとられるものとに分ける。

(d) 取引カードに収容する項目を決める。

はじめに、項目の種類を説明する。分類項目、参照項目、計算項目など。分類項目 (商品番号など) は、基本カードにも取引カードにも必要である。計算項目 (単価など) は、基本カードにあれば取引カードになくてもよい。

(e) 取引カードに収容する項目の配列の仕方を決める。

配列を決めるときは、次の点に注意が必要である。

関連カード上の位置

原書類の記載順序

紙カードのせん孔方法 (英字せん孔・数字せん孔)

項目の種類

(f) 項目の長さを決め、配列順序に従ってカード設計用紙に記入する (カード設計用紙は付録参照)。

紙カードの種類を簡単に見分けられるようにするために、次のような処理を行なっていること

を説明する。

- ・ コーナ・ カットの位置、角度を変える。
- ・ 紙カードの色を変える。
- ・ 紙カードの印刷色を変える。
- ・ せん孔する項目名を印刷する。

カード設計についての理解を深めるために、簡単な問題を与えて生徒に設計させるとよい。

(3) 原書類設計

原書類の設計については、次のことを説明する。

(a) カード設計にもとづいて原書類を設計する。

紙カードのせん孔順序と原書類の記載順序を一致させる。

(b) せん孔個所をわかりやすくする。

次のような注意が必要である。

- ・ せん孔個所を太線で囲む。
- ・ 項目名やせん孔行を明記する。

例

社 員 番 号			

18

21

(c) 原書類の種類をわかりやすくするために、色、大きさ、題字などで区別する。

(d) 原書類の管理のしやすさを考える。

総括表をつけて束(バッチ)として扱うことなど。

(e) せん孔カードから原書類をたどれるようにする。

紙カードに原書類番号や一連番号、バッチ番号などを記載するなどの方法を考える。

(4) 磁気テープのファイル設計

次のような点を説明して磁気テープ・ファイルの概念を理解させる。

(a) ボリューム・ラベル

ボリュームの概念とボリューム・ラベルの意味

(b) ファイル見出しラベル、ファイル終りラベル

(c) レコードの形式

レコード長、固定長レコードと可変長レコード

(d) ブロックの形式

物理レコードと論理レコード、ブロック化因数、可変長ブロックと固定長ブロック

(e) 最終ブロックの形式

パディング (padding)、パディング・レコードを含んだブロックと短ブロック (short block)

(f) レコードの長さとブロックの長さ

ボリューム・ラベルとファイル・ラベルについては、JIS規格が決まっているので、JIS規格に従って内容を説明することが望ましい。

磁気テープ・リール 1巻に収容できるデータ量について、第2部の磁気テープ記憶を復習しながら検討させてみてよい。たとえば、せん孔カード (カード・イメージのレコード) を何枚記憶できるか考えさせる。ブロックの大きさをえるることによって記憶できる個数が大きく変動することを理解させる。

演習問題 1

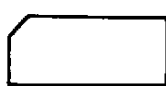
給与計算その他適当な問題を取上げて、次のような作業をさせる。

- (1) コード設計の必要な項目 (所属・社員番号など) を取出し、コードを決めさせる。コードを決めたら、なぜそのコードを決めたかを説明させ、有利な点、不利な点を議論させる。
- (2) 報告書・紙カード・原書類の順序でファイルの設計を行なわせる。設計の結果は、報告書設計用紙、カード設計用紙に記入させる。
- (3) 設計した結果について、有利な点、不利な点を検討させる。

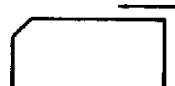
1.4 流れ図

第3部第1章で学んだ流れ図記号とその書き方を復習する。

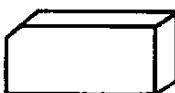
第3部で学んだもの以外に、JIS規格で決められている記号の中から、次のようなものを取上げて説明する。これらの記号は主としてシステム流れ図を書くときに使われる。テンプレートを与えてテンプレートの使い方を説明する (テンプレートは付録参照)。



紙カード



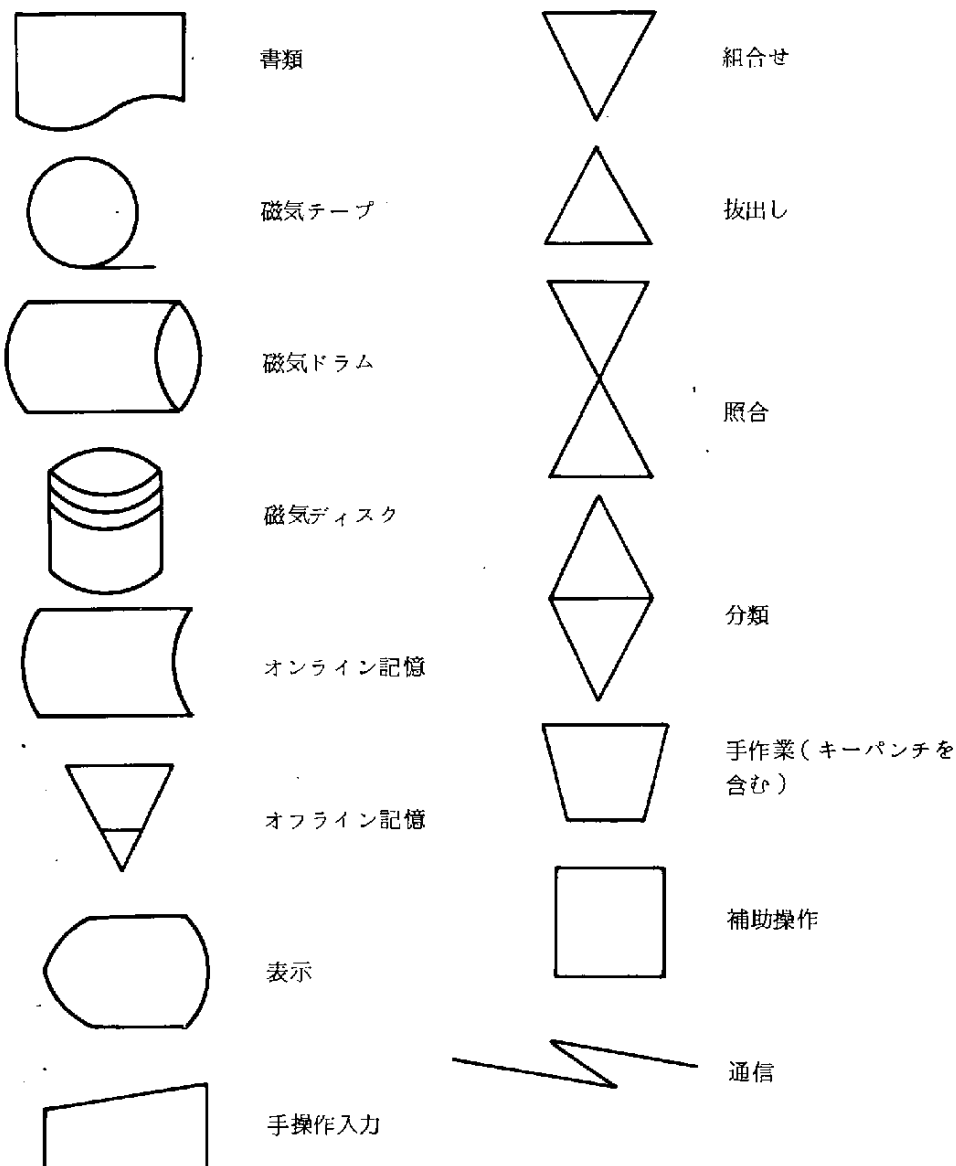
カードのファイル



カードの組



紙テープ



流れ図記号を使用して、システム流れ図を書く。次のことを説明する。

- ・システム流れ図によって作業の単位(run)が決り、相互の関係がはっきりする。
- ・作業の単位ごとにプログラムが必要である。
- ・システム流れ図を見ると、各単位ごとに使用されるプログラムとファイルがわかる。

1.5 プログラム設計

次のような点について説明する。

プログラム設計の手順

プログラム設計の手法

判断表

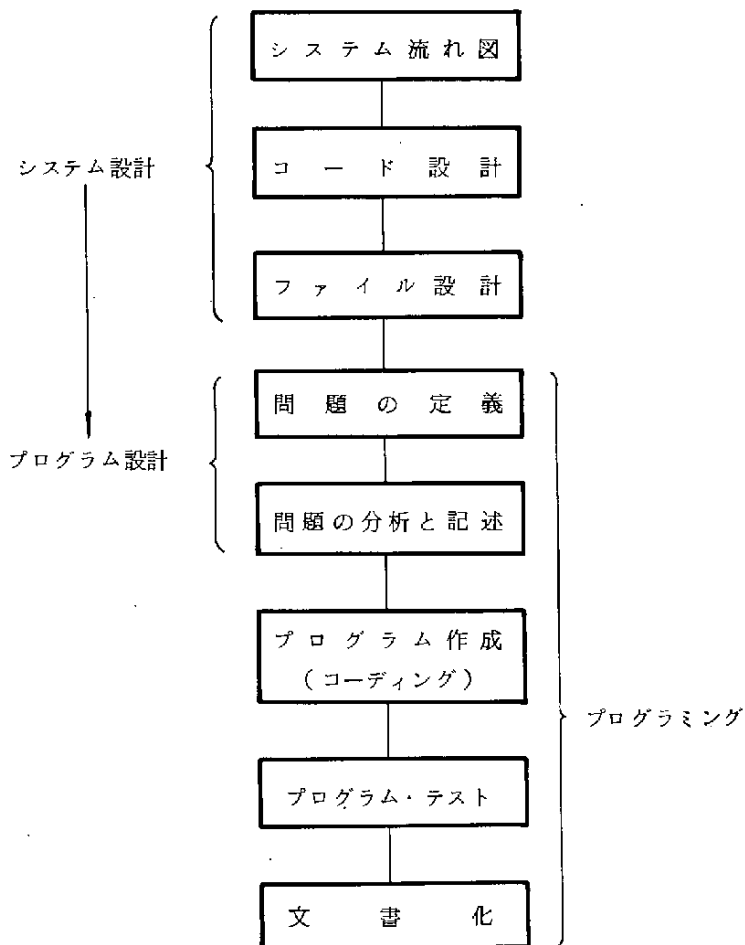
流れ図

(1) プログラム設計の手順

第3部第1章で簡単にプログラム流れ図の書き方を説明した。ここではもう少し詳しくプログラム設計の説明をする。

最初に、システム設計とプログラム設計の関連、プログラム設計の手順を説明する。

次のような図を示すとよい。



(2) プログラム設計の手法

(a) 問題の定義

問題の定義とは、問題を明確にすることである。プログラムを設計するに先だって、次のような点を明確にする。

入力ファイルの内容、形式

出力ファイルの内容、形式

プログラムに持たせるべき機能

これらは、システム設計書の中で記述されているので、プログラム作成者の立場からその内容を検討し、問題を明確にする。

(b) 問題の分析と記述

判断表(decision table)、プログラム流れ図(flowchart)などの概念を説明する。

判断表や流れ図を使ってプログラムを設計する手順を簡単に説明する。

(3) 判断表

判断表を使用する利点について説明する。

判断表の構成について述べる。次のような図を示して説明するとよい。

条 件	条件に対する答
行 動	行動をとるかどうかの指示

条件に対する答の記入の仕方(YかN)、行動をとる場合の記入の仕方(X)などを説明し、判断表の利用の仕方を習得させる。

判断表の書き方は、例をあげて説明した方がよい。例としては、たとえば次のようなものが考えられる。

例 座席の予約

規 則

		1	2	3	4	5	6	7	8
条 件	1 1等席申込み	Y	Y	Y	Y				
	2 2等席申込み					Y	Y	Y	Y
	3 1等席空席あり	Y	N	N	N		Y	N	
	4 2等席空席あり		Y	N		Y	N	N	N
	5 他の等席でもよい		Y	Y	N		Y	Y	N
行 動	1 1等席券発売	X					X		
	2 2等席券発売		X			X			
	3 1等席空席からマイナス1	X					X		
	4 2等席空席からマイナス1		X			X			
動	5 1等席待ち行列リストに書き加える			X	X			X	
	6 2等席待ち行列リストに書き加える			X				X	X

(4) プログラム流れ図

プログラム流れ図について、**粗い流れ図**(general flowchart)と**詳細流れ図**(detail flowchart)の説明をする。

例をあげて、流れ図の意義と流れ図作成手順を説明することが望ましい(流れ図で使用する用紙については、付録の流れ図用紙を参照)。

例として、次のような問題が考えられる。

「コンピュータ室へ、毎日売上伝票がカウンタ(売場)別に整理して送られてくる。それをもとに、各カウンタのその日の売上合計を計算して、売上日報を作成する」。

この問題について、システム流れ図とファイルの仕様(売上伝票・売上カード・売上日報など)を示し、プログラムの粗い流れ図と詳細流れ図を書く。

はじめに、システム流れ図を見て、入力・処理・出力という3つのステップの内容を明らかにすることが重要である。これが問題の定義にあたる。

この問題では、次のようになる。

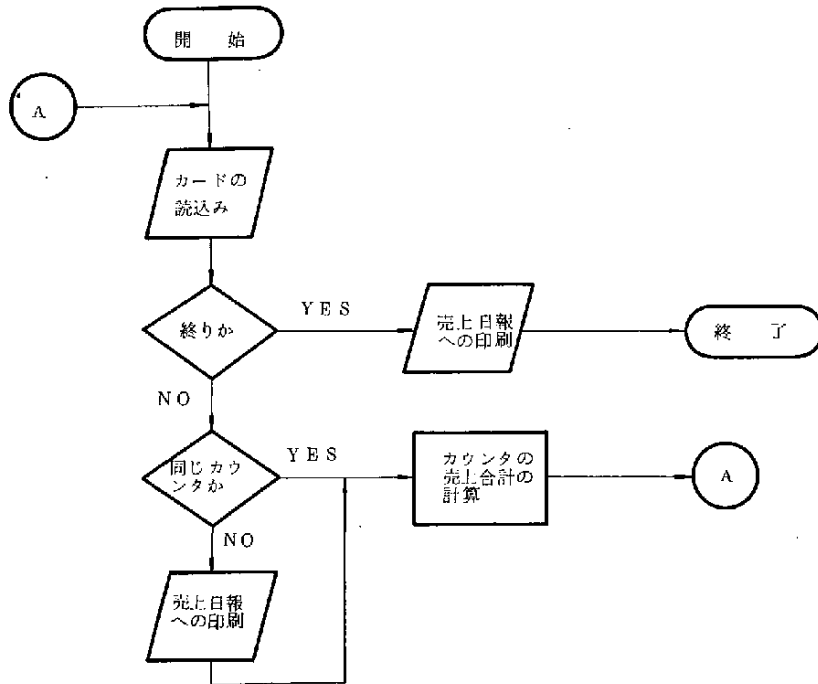
(a) カードの読込み(入力)

(b) カウンタの売上合計の計算(処理)

(c) 売上日報への印刷(出力)

これをもとに、粗い流れ図を作成する。

次のような程度のものでよい。



ここで流れ線の使い方を復習するとよい。流れ線の使い方には次のような約束がある。

- (1) 流れの方向は原則として、左から右へ、上から下へとする。
- (2) 流れの方向がこれに合わないときは、矢印を用いる。
- (3) 流れ線は交差してもよい。
- (4) 2つ以上の流れ線を集めて1つにしてもよい。

粗い流れ図をもとに、詳細流れ図を書く。詳細流れ図は、ルーチンやプロシージャごとに作成するので、ルーチンやプロシージャの意味、プログラムとの関係などを合わせて説明する。詳細流れ図にはすべての処理が正確に書かれていなければならない。

詳細流れ図は、文書(ドキュメント)としても重要な意味を持っていることを述べ、次のような点を説明する。

- (a) ボックスの中の表記の仕方
- (b) 分岐先のラベル
- (c) 原始プログラムとの対応づけ

演習問題2

実務の中から簡単な問題を取上げて、プログラム設計の演習を行なう。プログラム設計の基本的なことから身につけさせることが目的だから、条件はあまり複雑にしない方がよい。また実務に対する予備知識が必要なものは適切ではない。給与計算の例などは取上げやすいだろう。

プログラムの粗い流れ図、詳細流れ図を提出させたら、正確さだけでなく読みやすさという点からも評価し、どうしたら読みやすい流れ図を書くことができるかを生徒に考えさせてみるとよいであろう。

1.6 情報処理規格

情報処理の分野で標準化がなぜ必要なのか、標準化することによってどのような利点が考えられるか。現在、標準化はどのように進められているかなどを説明し、プログラミングの際に関連の深い情報処理規格を紹介する。

情報処理の分野の標準化は、多かれ少なかれ文書化と関連している。そこで文書（ドキュメント）を作る際に、標準化の考え方をどう生かしたらよいかについても説明する必要がある。

(1) 標準化の意義

標準化がなぜ必要か、標準化するとどのような利益があるかを説明する。例として、電力の周波数が関東の50ヘルツと関西の60ヘルツに分れていることによる不利益や、コンピュータが変ってプログラムやファイルを全部作りかえなければならなかったときの混乱などをあげるとよい。

標準化の利点を次のような点から説明する。

- (a) プログラムを開発するときの利点
- (b) コンピュータを使用するときの利点
- (c) 教育や情報交流のときの利点

標準化については、国際的な規格や国内の規格に頼るだけでなく、企業の中で、あるいは作業グループの中で、個々に決めて行かなくてはならないことも多い。

その際の問題点として、次のような点に触れることが望ましい。

(a) 計画的な標準化

標準化の時期や範囲を考える。

- (b) 国際規格や国内規格と矛盾しない。
- (c) 他の企業や関連部門の考え方と矛盾しない。

(2) 標準化の動向

次のような機関の活動に触れる。

- (a) 国際標準化機構 (International Organization for Standardization, ISO)
- (b) 通産省工業技術院日本工業標準調査会 (JIS規格)
- (c) 情報処理学会, その他の学会
- (d) (社) 日本電子工業振興協会
- (e) (財) 日本情報処理開発センター

(3) 情報処理規格の内容

情報処理に関係した規格を紹介する(付録参照)。特に次の規格については、プログラミングと非常に関係が深いので、内容を紹介し生徒の理解を深めておくことが必要である。

(a) 用語

情報処理用語

(b) 用紙および文書化

帳票の設計基準

規格票の様式

情報処理用流れ図記号

(c) 媒体

情報交換用磁気テープのラベルとファイル構成

(d) 電子計算機用プログラム言語

F O R T R A N 水準 7 0 0 0, 5 0 0 0, 3 0 0 0

(e) コード

日付の表示

時刻の表示

都道府県コード

市区町村コード

性別コード

指導上の留意点

- (1) システム設計の手順は、事務についての実際的な知識があると理解しやすい。したがって

生徒が事務についての知識に乏しいときには、実際の事務処理に触れさせるようなくふうが必要である。

システムについて理解させるときに、身近な例を取上げると有効である。たとえば、次のようなものが考えられる。

(例) 洋服の仕立て

特定の人に合わせて洋服を作る作業を適用業務、ミシンをハードウェア、裁縫技術をソフトウェアにたとえる。

洋服屋は仮縫いや仕立ての手順を組んだり、1枚の布からできるだけ多くの服地をとることを考えたりする。それがシステム設計の中の流れ図を作ったり、ファイル設計をする作業にあたるといえる。

(2) コード設計やファイル設計では、実習を行なって知識や技術を身につけさせることが必要である。実習の結果を生徒自身に評価させて、問題点を出しあうことがよい。議論したり討論した内容は、実際の場で生かされるに違いない。

(3) 流れ図記号については、知識として身につけても意味がないので、生徒が実際にプログラムを作るときに、流れ図規格に従って流れ図を書くように、また流れ図を書き、流れ図に従ってコーディングをするように指導することが重要である。

(4) プログラム設計では、実習を行なってプログラム設計の基礎的なことを身につけさせる。

(5) 情報処理規格については、プログラム作成と文書化の関係について、できるだけ時間をかけて説明することが望ましい。規格の中には、文書化と直接関係しない規格もあるが、ほとんどのものは、プログラミングの過程で作業の結果を文書にまとめるときに意味を持つものである。プログラムを作成するときの心構えとして、標準化に対する姿勢を身につけさせるよう指導する必要がある。

第2章 コンピュータの応用

この章では次の用語を教える。

経営情報システム, 情報検索, リアルタイム・システム, オンライン・システム, 座席予約, オンライン・バンキング・システム, メッセージ交換システム, データ・ベース, コントロール・システム

目 標

コンピュータがどう利用されているかを説明し、コンピュータの応用について理解させる。コンピュータの利用の仕方は、コンピュータを利用する業務（適用業務）によって非常に異なる。したがって生徒に関係の深い分野を選び、その分野でコンピュータがどう使われているかを説明するのが望ましい。

少なくとも次のどれかの分野に焦点をしばって説明する。

- (1) 製造工業
- (2) 流通
- (3) 新聞、放送、出版
- (4) 金融
- (5) 電気、ガス
- (6) 官公庁

内 容

特定の適用分野を選んで、その分野のコンピュータ利用について説明する。次のような点を説明する。

- ・ コンピュータ利用の歴史
- ・ コンピュータ利用の現状
- ・ コンピュータ応用の技法

それぞれについて説明すべき内容を述べる。

2.1 コンピュータ利用の歴史

コンピュータがいつから使われ始め、コンピュータ利用がどのように発展して来たかを説明する。生徒が1企業内に限られている場合には、その企業におけるコンピュータ利用の歴史を述べてもよい、企業内の歴史を述べるときには、次のような点に触れると有効である。

- (1) コンピュータを利用した動機
- (2) 適用業務
- (3) コンピュータ利用による変化

事務手続き、人の移動、業務形態などの変化に触れる。

- (4) コンピュータ利用の効果

2.2 コンピュータ利用の現状

自社および同じ分野の企業で、コンピュータがどう使われているかを述べる。具体的な業務を取上げて、その業務を機械化した場合の目標や効果を説明するとよい。

コンピュータの利用によって機械化された部門の処理がどうなっているか、機械化に対してどのような評価がされているかについても理解させたい。機械化された部門を見学したり、機械化を推進した人を招いて聞いたりする方法も考えられる。

2.3 コンピュータ応用の技法

コンピュータ応用の技法として、次のような内容について簡単に説明する。

- (1) 経営情報システム

内容にあまり立入る必要はない。

考え方を説明する。

- (2) オンライン・システム、リアルタイム・システム

座席予約、オンライン・バンキング・システム、メッセージ交換システムなどについて、概要を説明する。

- (3) データ・ベースを活用した情報検索システム

特許情報サービスや、文献検索などの例をあげて説明する。

- (4) コントロール・システム

化学や製鉄のプロセス制御、航空管制、道路交通制御などの例をあげて説明する。

- (5) 人とコンピュータとの対話システム

コンピュータによる設計やコンピュータによる教育などの例を紹介する。

指導上の留意点

それぞれの部門の適用業務を取上げるときは、次にあげる各業務の代表的な適用業務を参考にするとよい。

概括的な説明をする場合には、それぞれの分野の主な適用業務について、全体的な説明をする。

概括的な説明に加え、生徒に最もなじみの深い適用業務を1, 2取上げて選び、コンピュータの応用について具体的な説明をする。

各業種の代表的な適用業務

製造業 生産管理・工程管理・部品展開・作業管理・品質管理など。

流通 市場分析・商品情報検索・販売管理・仕入管理・セールスマン管理・手形管理・契約事務など。

官公庁 経済分析・計量経済モデル・職業紹介・恩給支給・住民登録・住民税・固定資産税・水道料金・国民健保など。

金融 預金業務・貸付業務・融資統計・利息計算・手形割引・為替業務・貸付信託・株式など。

新聞、放送、出版 広告業務・購読料請求・編集業務・スポンサー管理・視聴率計算など。

電気、ガス 給電運用・系統運用・配電・配電工事管理・電灯電力料金計算・火力発電技術計算
ガス料金請求・ガス工事精算・容器管理など。

一般的な業務 経理業務・給与計算・人事統計・在庫管理・資材管理・減価償却・販売統計・売掛金管理など。

特定の業務に対するコンピュータ利用の問題を説明するとき、どの程度まで立入ったらよいかについて在庫管理の例を上げておこう。

在庫管理に対するコンピュータの応用

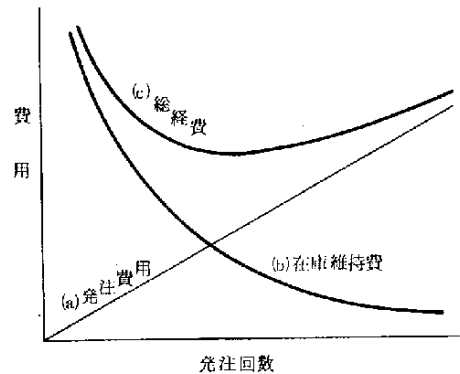
在庫管理の必要性和在庫管理の要素（発注点、発注量など）を説明する。

静的な、変動（傾向変動、季節変動など）の少ない在庫モデルを示し、代表的な発注の仕方を説明する。それには、定期発注と定量発注とを取上げればよい。定期発注では、発注サイクルは一定、発注量は不定になる。定量発注では、逆に発注サイクルは不定、発注量は一定になることを説明する。

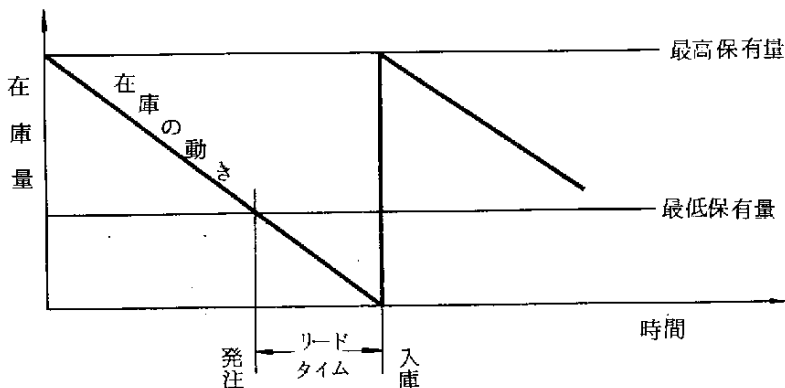
定期発注よりも定量発注の場合について詳しく説明した方が、在庫管理の問題を明らかにしやすいであろう。その1つの問題として、発注量をいくらにすると経費が最小になるかを考えさせる。

これらの問題には、種々の要素が関係しているから、ある程度要因を単純化または選別することが大切である。

たとえば右図のように、発注費用と在庫維持費とが等しくなるところで、総経費は最小になることに注意して、経費が最小となる発注量（経済発注量）を求めさせる。経済発注量による在庫管理を手作業で行なう場合の2ピン法について説明する。



これをコンピュータで処理する場合の方法を説明する。リード・タイム（発注から入庫までの待ち時間）中の平均出庫量を最低保有量とし、最低保有量プラス経済発注量を最高保有量とする。在庫が減少して最低保有量を割ると、最高保有量に達する量を発注するのである。



その処理手順は、次のようになる。

- (1) 品目ごとの在庫マスタ・ファイルを作る（品目番号、品目名、リード・タイム、在庫残高、最高保有量、最低保有量などを記録）。
- (2) 出庫のたびに在庫ファイルを作り、マスタの在庫残高からマイナスする。
- (3) 在庫残高が最低保有量以下かどうかをテストし、以下であれば発注ファイルを作る。
- (4) 入庫があれば入庫ファイルを作り、マスタの在庫残高にプラスする。

このような方法を自動発注方式という。

以上のように、静的な在庫モデルでは需要を一定と考え、最低保有量を一定としている。

需要の変動を前提として在庫管理を行なうことは、相当高度な技法を要するが、そのあらましについて説明すれば、コンピュータをこの分野に応用する意義を明確にすることができるであろう。

参考文献

1. "MODERN DATA PROCESSING"

R. R. ARNOLD

JOHN WILEY & SONS・INC.

昭44 3,580円

2. 電子計算機ユーザ調査年報 1970年版

情報処理学会編

日本経営科学研究所

昭45.3 880円

3. 株式会社日立製作所発行のもの

EDP導入マニュアル

計画から稼働まで

昭44 実費

業務別アプリケーションI

給与、人事

昭44 実費

業務別アプリケーションIV

購買管理

昭44 実費

9040預金システム

昭44 実費

4. 富士通株式会社発行のもの

システム分析と設計の進め方

昭44 実費

プログラミング計画と標準化

昭44 実費

FACOMによる生産管理(1)

部品展開

昭44 実費

FACOMによる人事業務

昭44 実費 など。

5. 日本電気株式会社発行のもの

FICS, ISIM説明書

需要予測在庫管理

昭44 実費

予算の編成とシミュレーション

原価計算・経理・予算編成

昭44 実費

FACTOR概説書

生産計画(注文処理・部品展開・日程計画等)

実費

6. 日本アイ・ビー・エム発行のもの

電子計算組織導入計画概説書

システム設計・プログラムの準備・標準化

昭40 実費

基本的なシステム検討の手引

昭38 実費

第2章 現行事業の理解

第3章 システム要件の決定

第4章 新システムの設計

7. EDPシステム設計

土岐秀雄

日本経営出版会

昭44 1,500円

実例的な事例を多く含んでいる。

8. コードの手引き

— データ・コードの標準化と編成 —

情報処理学会

昭43.10 700円

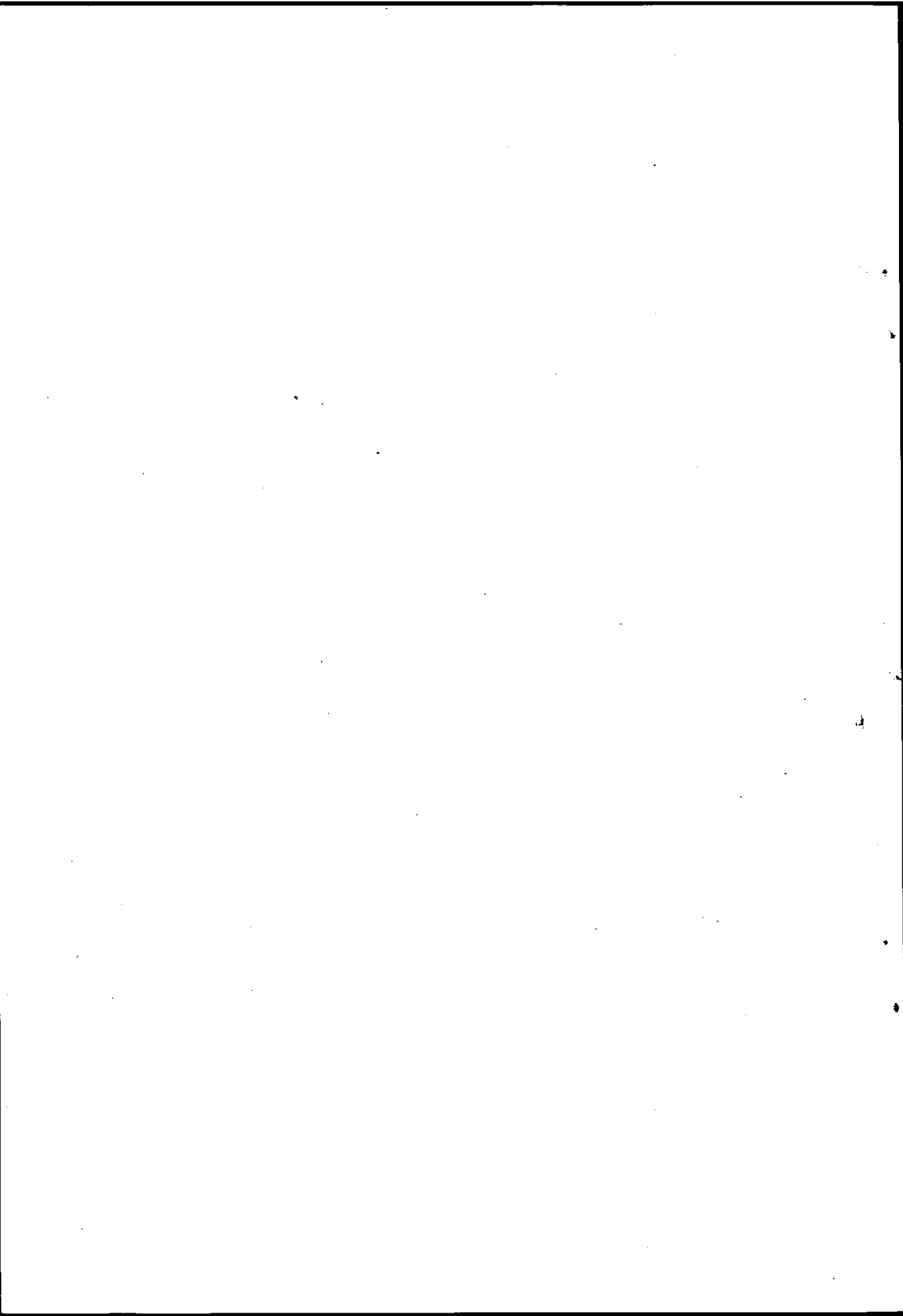
第2章 データ・コードの体系化

9. コード化の手引き

日本アイ・ビー・エム

昭44 実費

實 習 編



第 1 部 プログラミング (1) FORTRAN

次の用語を教える。

エラー・メッセージ, 机上デバグ, けたあふれ, 関数, 算術式, 書式, ステートメント(文),
定数, テスト・データ, 配列, 引数, 変数, ループ

教育の目標

目標は JIS FORTRAN 水準 3000 を習得し, 与えられた問題を FORTRAN で記述できるようにすることである。

そのために次のようなことが必要である。

- (1) FORTRAN の文法を習得する。
- (2) 適宜機械実習を行ない, FORTRAN でいくつかプログラムを組ませる。

JIS FORTRAN 水準 3000 は基本的な FORTRAN である。基本的な FORTRAN が習得できれば, 実務について, それぞれのコンピュータのレベルに合わせた FORTRAN の文法が必要になったとき, それを独力で習得できるはずである。

内容のあらまし

JIS FORTRAN 水準 3000 の内容を次のように分類して教える。

内 容	標準講義時間 (単位:時間)	機 械 実 習 (標準)	機 械 実 習 (最低)
FORTRAN 概論	2		
FORTRAN プログラムの形式			
○コーディング・シートとコーディング・シートへ記入するときの注意	1		
○FORTRAN で使用できる文字	1		
○行(注釈行, END 行, 開始行, 継続行)	1		
○文	1		
○文番号	1		
○英字名	2		

内 容	標準講義時間 (単位:時間)	機 械 実 習 (標準)	機 械 実 習 (最低)
データの型 ○定 数 ○変 数 ○配 列	2.5 3 5		
基本的な文(1) ○算術代入文 ○GO TO文と算術IF文 ○入出力文(1)	5 6 7		
演習問題(1)		10	0
基本的な文(2) ○DIMENSION文 ○DO文とCONTINUE文 ○添字式 ○STOP文とPAUSE文 ○入出力文(2)	4 7 3 0.5 11		
演習問題 (2)		10	5
基本外部関数と組み関数	7		
演習問題(3)		13	0
文関数	7		
演習問題(4)		15	8
副プログラム ○関数副プログラム ○サブルーチン副プログラム	9 9		
演習問題(5)		15	0
COMMON文とEQUIVALENCE文 ○COMMON文 ○EQUIVALENCE文	6 6		
演習問題(6)		20	8
書式なし入出力文	4		

内 容	標準講義時間 (単位:時間)	機 械 実 習 (標 準)	機 械 実 習 (最 低)
補助入出力文	2		
演習問題(7)		20	10
ま と め	4		
合 計	117.0	103.0	31

機械実習は、与えられた問題を分析し、流れ図を書く段階から、コンピュータにかけて問題を解決するまでの時間を含めたものである。実際には、それぞれの環境によってコンピュータを使用できる時間は異なるので、最小限必要だと思われる時間数も載せた。

指導上の留意点

同じFORTRANでもコンピュータによって文法の異なる点があるから、機械実習に先だって使用するコンピュータのFORTRAN説明書を参照して、JIS FORTRAN 水準3000との関係をつかんでおく必要がある。もしそのFORTRANが水準3000を下まわっている場合には、混乱を生じないために、水準3000との相異点を明示しなければならない。機械実習にあたってジョブ制御カードが必要な場合には、教師が用意して生徒に渡し、その使い方を指示する。ジョブ制御カードの目的を知りたがる生徒もいると思われるので、簡単に説明するのもよい。しかし、最初からジョブ制御カードの説明をしても混乱を招くだろうから、演習問題2のときにするのがよいだろう。

コンピュータを使用する場合、教師が生徒のエラーを発見したときの対し方には、慎重な配慮が必要である。エラーを発見して、それをただちに指摘することは、必ずしも好ましい結果をもたらさない。コーディングをした後、それを最初にコンピュータに通すときには、よほどコンピュータ使用時間に制約があるのでない限り、教師は生徒の間違いを指摘すべきではない。デバグの仕方も実習時に指導しなければならないが、これも演習問題2のときにしたい。

それぞれの演習問題についての指導上の留意点は、演習問題のところで述べる。この実習編を通じて、相互参照テーブル(cross reference table)については、触れないでおいた方がよいだろう。またコンピュータを使用しないときでも、できるだけ多くの演習問題を与えるようにして生徒に自主的に解かせてみる必要がある。

問題を独力で解いてみるのが、FORTRANの文法を理解する上では有効だと思われるが、コンピュータの使用時間に制約がある場合には、共同作業も考えられる。この場合、グループの人数は多くても4人ぐらいまでである。少し複雑な問題を扱うときには、グループ内の各人に担当部分を分担させて、それぞれに責任を持たせた形で共同作業をするという方法もある。共同作業で注意しなければならないのは、同一の問題を共同で考えるという方法では教育効果がないということである。共同で考えるという方法をとると、主に考えるものとそれに従うものができて、限られた人しか力が出ないということになりがちだからである。

1. FORTRAN 概論

目 標

FORTRANの特徴、歴史、FORTRANで書かれたプログラムが実行されるまでの過程などを理解させる。基礎編第3部第3章言語のあらましの復習を兼ねる。

内 容

FORTRANの特徴では、科学技術計算向きであることを述べる。COBOLの特徴と対比させると理解しやすいだろう。

FORTRANの歴史では、FORTRANがどのようにして生れ、世界中で使用されるようになったかを説明する。FORTRANの文法にも機械によって多少の違いがあるので、これから習得しようとしているFORTRANは、大体においてどのコンピュータにも使える基本的な部分であることを話す。

FORTRANで書かれたプログラムが実行されるまでの過程では、われわれが書いたプログラムをコンピュータが直接実行してくれるのではなく、中間に機械語に翻訳するという作業があり、機械語に翻訳されたプログラムによって、はじめて実行が行なわれることを理解させる。

2. FORTRAN プログラム形式

目 標

おおまかに、FORTRAN プログラムとはどんなものかということを理解させる。

内 容

次のような説明をする。

- (1) コーディング・シートと、コーディング・シートへ記入するときの注意
- (2) FORTRANで利用できる文字
- (3) 行(line), 注釈行(comment line), END行, 開始行(initial line), 継続行(continuation line)
- (4) 文(statement)
- (5) 文番号(statement label)
- (6) 英字名(symbolic name)

2.1 コーディング・シートとコーディング・シートへ記入するときの注意

FORTRANのコーディング・シートを示して、けたの説明をする(コーディング・シートについては付録を参照)。

2.2 FORTRANで利用できる文字

FORTRANで利用できる文字を理解させる。せん孔ミスを防ぐために、コーディング・シートに書く文字の書き方を決めておく必要があることを説明する。

2.3 行

注釈行, END行, 開始行, 継続行の説明をする。継続行を教えると、注釈行も継続できるように誤解する生徒がいるので、注釈行は継続できないことを徹底させる。

2.4 文(ステートメント)

1つの文は開始行だけか、または継続行のあるときは、それを続けたものであるということ、継続行の第7けたは、直前の行の第72けたから続いていることを理解させる。

2.5 文 番 号

文には数字の文番号を付けることができるが、同じ番号を2つ以上の文に付けることはできないことを理解させる。

2.6 英 字 名

英字名は5文字以内の英数字で、第1字目は必ず英字でなければならないことを徹底させる。練習問題を課してみるとよい。

指導上の留意点

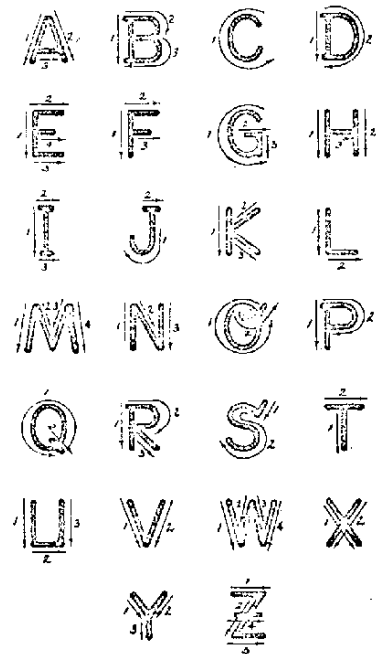
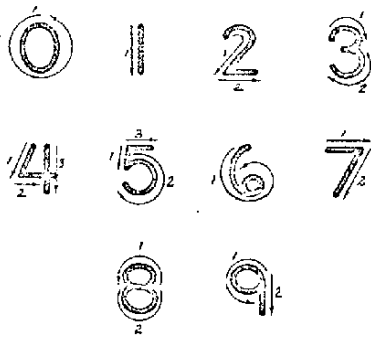
簡単なプログラムを示して説明するとよい。例としては次のようなものが考えられる。

「2つの数をカードから読んで、和、差、積、商を計算して印刷する」。

```
0      EXAMPLE
      READ(5,10)A,B
10     FORMAT(2F5.1)
      S=A+B
      D=A-B
      P=A*B
      Q=A/B
      WRITE(6,20)A,B,S,D,P,Q
20     FORMAT(1Hb,6F10.1)
      STOP
      END
```

コーディング・シートに記入する文字については、演習を始める前に次の例を参考にして適当に規則を決めておく。

アメリカ規格案



一般的な書き方

0, Ø (ゼロ)	Ø (オー)	1 (イチ)	I, i (アイ)
/ (スラッシュ)	Z (ゼット)	D (ディ)	

3. データの型

目 標

実数型、整数型という2種類のデータの型について理解させる。

内 容

3.1 定数

整数定数 (integer constant), 実定数 (real constant) の説明をする。特に指数部を持つ定数 (constant) の表わし方を習得させる。1つの数値もEを使って表わすのに、いろいろな方法があることを理解させ、正規化された数の概念をはっきりさせる。機械実習をするコンピュータのもつ整数、実数それぞれの有効けた数を教えておくとよい。

3.2 変数

変数 (variable) とは結局は記憶場所に付けられた名前であり、そこに整数型のデータが入れば 整数変数 (integer variable) であり、実数型のデータが入れば 実変数 (real variable) であることを認識させる。実数型変数、整数型変数を表わすときの規則を習得させる。

3.3 配列

配列 (array) の概念を理解させるために、配列、配列要素 (array element), 添字 (subscript) の説明をする。生徒がベクトルや行列の知識を持っているときは、これに例にとれば理解しやすいと思われる。

次のことを理解させる。

- (1) 配列を利用する利点
- (2) 実数型と整数型の配列
- (3) 添字には整数変数が整数を書く。ここでは添字式には触れない。
- (4) 負や0の添字は許されない。

4. 基本的な文(1)

目 標

次のようなFORTRANの実行文と非実行文について理解させる。

I 実行文(executable statement)

(1) 代入文(assignment statement)

算術代入文(arithmetic statement)

(2) 制御文

算術IF文(arithmetic IF statement)

GO TO文

CALL文

CONTINUE文

STOP文とPAUSE文

DO文

RETURN文

(3) 入出力文(input/output statement)

READ文, WRITE文

補助入出力文(auxiliary input/output statement)

II 非実行文(non-executable statement)

(1) 宣言文(specification statement)

DIMENSION文, COMMON文

EQUIVALENCE文

(2) FORMAT文

(3) 文関数定義文(statement function defining statement)

(4) 副プログラム文(subprogram statement)

FUNCTION文

SUBROUTINE文

これらの文を、簡単なものから複雑なものへと順に理解させたい。

内 容

4.1 算術代入文

次の順序で指導する。

- (1) 算術代入文の簡単な例を示し、その意味を理解させる。

例としてあげるものは、 $A = C + D$ 、 $A = C - D$ の程度でよい。ここで注意すべきことは、数学の等号と算術代入文の等号とは意味の違うことである。等号の左辺は1個の変数名であり、右辺は算術式であることを理解させる。

- (2) 算術演算子 (arithmetic operator) ($+$, $-$, $*$, $/$, $**$) と、かっこの使用の仕方について説明し、演算の優先順位を理解させ、少し複雑な数式を書表わせるようにする。教師が黒板などに書いた数式を、算術代入文で書かせてみる。

- (3) 算術式は、次の規則に合っていないなければならないことを説明する。

算術演算子 $+$, $-$, $*$, $/$ の		演 算 結 果
右 側	左 側	
実 数 型	実 数 型	実 数 型
実 数 型	整 数 型	許されない
整 数 型	整 数 型	整 数 型
整 数 型	実 数 型	許されない

算術演算子 $**$ の		演 算 結 果
左側 (底)	右側 (指数)	
実 数 型	実 数 型	実 数 型
実 数 型	整 数 型	実 数 型
整 数 型	整 数 型	整 数 型
整 数 型	実 数 型	許されない

実際に数を代入して演算結果がどのようなになるかを説明する。特に整数を整数で割る演算では、結果は小数点以下切捨ての整数の形で求められるので、平方根を求めたいときに、次のような間違いを犯さないように注意する。

$$Z = (A + B) ** (1 / 2)$$

これは、算術式の規則には合っているが、 $1 / 2$ は0になってしまう。

(4) 次のように算術代入文の右辺と左辺の型が異なる場合、

(a) 実数型=整数型

(b) 整数型=実数型

それぞれの結果がどのようなになるかを具体例で説明する。さらに、 $A = B$ や、 $PAI = 3.1415$ など算術式が1つだけの変数や定数から成る場合があることにも触れる。

4.2 GO TO文と算術IF文

(1) GO TO文

無条件GO TO文と計算型GO TO文を説明する。

(2) 算術IF文

算術IF文を説明する。かっこの中に算術式を書くことができることを教え、また、分岐先の3個の文の番号のうち2個が同じであってもよいことを理解させる。

4.3 入出力文(1)

書式つき入出力文について説明する。FORMAT文と対にして理解させるが、ここでは簡単なものについて理解できれば十分であり、複雑な入出力文は後にする。配列ヘデータを読んだり、配列にはいつている数値を印刷する方法の説明や、FORMAT文のかっこの中にかっこや/を含むものは除く。

(1) 書式つきREAD文(formatted READ statement) (1)

カードにせん孔された数の読み方を説明する。FORMAT文では、F変換、I変換、E変換、X変換を説明する。

F変換とE変換では、カード上に小数点が打たれている場合と、そうでない場合の違いを明らかにする。

(2) 書式つきWRITE文(formatted WRITE statement) (1)

数字や文字を印刷させる方法を教える。上述(1)の変換に加えて、H変換を説明する。行送り制御を説明し、WRITE文に対するFORMAT文には、行送り制御を書く習慣をつけさせるのが望ましい。

5. 演習問題 (1)

目 標

いままでに習得したことを利用して簡単なプログラムを書く演習を行なう。

内 容

次のような問題、またはそれと同程度のレベルの問題を与える。

(1) 体積、表面積、面積などを計算させる問題

(2) 2次方程式 $ax^2 + bx + c = 0$ の根

$$x = \frac{-b \pm (b^2 - 4ac)^{\frac{1}{2}}}{2a}$$

を求める問題

データは算術代入文で与えてもよいが、カードから読込ませてもよい。プログラムを終らせる条件として、データの最後に特殊な数を入れるか、データの個数を決めるかして、算術IF文で判定させる。

指導上の留意点

円の面積 $S = \pi R^2$ と、円周 $T = 2\pi R$ を計算することを例にとり、指導上の留意点を述べてみよう。最終カードとしては、半径0のカードを与える。プログラムは次のようになる。

```
20  READ(5,101)R
101  FORMAT(F12.2)
      IF(R)30,30,40
40   S=R**2*3.14
      T=R*2.0*3.14
      WRITE(6,102)R,T,S
102  FORMAT(1H0,7H HANKEI=,F12.2,10X,6H HENSHU=,F15.2,
1      8H MENSEKI=,F15.2)
      GO TO 20
```

30 STOP

END

次のような点を考慮しながら指導することが望ましい。

- (a) 流れ図を書かせる。その際手順さえはっきりつかめているならば、流れ図が完全に書けるようになるのは少し後になっても、とがめない（流れ図記入用紙については付録を参照）。
- (b) コーディングのときに注釈行を適宜そう入させる。この場合プログラムの最初には、プログラマの名前を入れた注釈行を入れさせる。これは、だれの書いたプログラムかをはっきりさせるのに有効である。また、あとで整理しやすいように、72けたから80けたの間に、カードの順序を示す数字を書かせるようにするとよい。数字は、あとで途中にそう入できるように10、20、30……のように数を飛ばしてつける。
- (c) 初めての機械実習であるから、データの書式や、印刷の書式は教師側で与えてよいが、その書式に合ったデータ・カードは生徒に作らせるようにする。実数型のデータを扱う場合には、小数点をデータ・カードにせん孔しなくてもよい。
- (d) FORMAT文は、対応するREAD文、WRITE文のすぐ後に書かせるようにする。
- (e) ジョブ制御言語は教師が用意をして、カードのそう入場所を指示する。できることならば、生徒の作成したプログラムのカードとは色を別にする。
- (f) カードのせん孔を生徒がしなければならない環境のところもあるだろうし、また実際にデバッグの際カードの修正などの必要がおこるので、カードのせん孔の仕方を指導する。このとき、いなくなったカードの捨て場所などを指示して、整頓する習慣をつけさせる。
- (g) コンパイル・エラーがあるとき、その見つけ方、エラー・メッセージの見方、訂正の仕方を指導する。カードを入替える場所をはっきりさせるために、その位置にあたるカードを逆にするなどの方法を教える。
- (h) 実行時のエラーも、なるべく生徒自身に発見させたいが、生徒から質問を受けたらそれに対して適切な助言を与えたい。コンピュータの使用時間が少ないときはもちろんだが、十分にあるときでも、机上デバッグをていねいにして、機械を有効に使うようにさせる。
- (i) 最後に一通り結果が出たら、どのような解法を用いたかを発表させ、1つの問題に対する解法は何通りもあることを理解させる。たとえば、この例ならば、 $P = 3 \cdot 14$ という文をREAD文の前に入れて、算術代入文にPを使用する方法もある。また、答は正しくても、途中でむだなことをしているようなときには、そのことを指摘して、なぜむだなのかを理解させる。

6. 基本的な文(2)

目 標

DIMENSION文、DO文などを教え、文についての理解を深める。

内 容

6.1 DIMENSION文

DIMENSION文を書く場所とDIMENSIONを宣言したときに記憶場所がとられる順番を習得させる。特に2次元の配列の場合について説明する。

6.2 DO文とCONTINUE文

DOを使った繰返しの処理(DOループ)について説明する。

(1) DO n $i = m_1, m_2$ と

DO n $i = m_1, m_2, m_3$ の説明をする。

(2) DOの端末文(DO terminal statement), CONTINUE文を習得させる。DOの端末文として禁止されている文に触れる。

(3) DOの範囲からGO TO文, 算術IF文を使ってDOの範囲外に出るときの制御変数の値と, DOが満足された後の制御変数の値がどうなるかを理解させる。

(4) DOの範囲内での制御変数の再定義の禁止

(5) DOの入れ子(nest)

DOの重ね方の説明と, DOの範囲外からDOの範囲にはいることは禁止されていることを理解させる。

6.3 添字式

添字として式が使えること, その式の形などを説明する。

6.4 STOP文とPAUSE文

STOP文とPAUSE文を説明する。

6.5 入出力文(2)

(I) 書式つきREAD文(2)

配列への数値の読ませ方を習得させる。

次のそれぞれの命令を説明する。

- (a) (i) DIMENSION A(10),B(10)
 READ(5,10)A,B
 10 FORMAT(.....)
- (ii) DIMENSION A(10),B(10)
 DO 30 I=1,10
 READ(5,10) A(I),B(I)
 10 FORMAT(.....)
 30 CONTINUE
- (iii) DIMENSION A(10),B(10)
 READ(5,10)(A(I),I=1,10),(B(I),I=1,10)
 10 FORMAT(.....)
- (iv) DIMENSION A(10),B(10)
 READ(5,10)(A(I),B(I),I=1,10)
 10 FORMAT(.....)
- (b) (i) DIMENSION A(2,3)
 READ(5,20) A
 20 FORMAT(.....)
- (ii) DIMENSION A(2,3)
 READ(5,20)((A(I,J),J=1,3),I=1,2)
 20 FORMAT(.....)
- (iii) DIMENSION A(2,3)
 READ(5,20)((A(I,J),I=1,2),J=1,3)
 20 FORMAT(.....)
- (iv) DIMENSION A(2,3)
 DO 30 I=1,2
 READ(5,20)(A(I,J),J=1,3)

```

20  FORMAT ( ..... )
30  CONTINUE

(c)  DIMENSION A( 2, 3 )
      DO 30  I = 1, 2
      DO 30  J = 1, 3
      READ( 5, 20) A(I, J)

20  FORMAT ( ..... )
30  CONTINUE

```

FORMAT文の説明では／とカッコを含んだものも付け加える。入出力並びとFORMAT文の欄記述子が1対1に対応していない場合は、どのようなになるかも説明する。

(2) 書式つきWRITE文(2)

(1)で述べたことと同様の内容を習得させる。

2つ以上のREAD, WRITE文で1つのFORMAT文を参照可能なこと, FORMAT文の置ける位置の説明をする。

指導上の留意点

ここで基本的な文の説明は終るので、いくつかの例をあげて、配列と普通の変数の混じったものなど複雑な方面へ話題を進めていく。ただし、あまり複雑なREAD文, FORMAT文を書くと、つまらない間違いを犯しやすいので、分割可能なものは分割する習慣をつけさせたい。

7. 演習問題 (2)

目 標

以上のことを習得すれば、相当複雑なプログラムも書けるはずである。学んだ知識を演習問題の中で活用させてみよう。

内 容

演習問題として次のようなものを取上げる。

- (1) 数を大きい順、小さい順に並べ替える。
- (2) いくつかの数の平均、標準偏差などを求める（平方根は0.5乗する）。
- (3) 多項式の値を求める。
- (4) 素数を求める。
- (5) 行列の加算、乗算など。

指導上の留意点

1つの問題に対する解法は何通りがあるだろう。機械実習が終わった後でいくつかの解法を示し、それぞれの特徴を指摘する。生徒に解法を発表させるのもよい。

(1) はよく使われる練習問題である。 $a_1, a_2, \dots, a_n$ を大きい順に並べる場合、よく採用される解法は、 a_1 と a_2 を比べ $a_1 \geq a_2$ ならそのまま、 $a_1 < a_2$ なら数の入替えを行ない、次に a_2 と a_3 を比べるということを繰返すものである。ここで問題になるのは、この処理を何回繰返したら数が大きい順に並ぶかということである。大きい順に並んだことを確認するには、スイッチをもうけておき、入替えがおきたらそれをONにして、一通りの処理をしたあと、スイッチの状態を調べるとか、常に最大の繰返し回数である $(n-1)$ 回の処理を行なうなどの方法をとればよい。

(2) 平均の計算で、結果を実数で出したいときには、割る数を実数型に変換して、実数/実数の計算をしなければならないことに気づかせる。

(3) 多項式 $f(x) = a_1 x^n + a_2 x^{n-1} + \dots + a_n x + a_{n+1}$ において、次数 n と係数 $a_1, a_2, \dots, a_{n+1}$ をカードから読ませる。 x はカードから読ませてもよいが、プログラムの中

で与えてもよい。 x に初期値、きざみの幅、最終値を与えて関数表などを作成させてみてよい。

多項式 $f(x)$ を $f(x)=(\dots(((a_1x+a_2)x+a_3)x+a_4)x+a_5)x+\dots+a_n)x+a_{n+1}$ のように変形して計算させる解法が便利であることを理解させる。

(4) ある数 N が素数かどうかを判定する方法は、 $N>M$ になる M で割切れるかどうかということである。 M の選び方にも問題があるが、 M で割切れるということは $N/M \times M$ と N とが等しいことである。

(5) 行列の演算は、FORTRANを学ぶときの予備知識として知っておいた方がよいので、もし生徒に知識のないものがいたら教えることにする。

FORTRANプログラムをコンピュータに通すときに使用する数枚のジョブ制御カードの意味を簡単に説明し、理解させる。

データ・カードの設計は、すべて生徒に自由に行なわせ、それに合わせたFORMAT文を書けるようにさせる。印刷の書式は、教師側で指定して、それに合わせたFORMAT文を書かせればよい。

コンピュータを使ってデバッグをする方法を指導する。使用するコンピュータのマニュアルに、今の段階で教えても理解しやすいと思われるデバッグ文があれば、それを教えるのもよいが、WRITE文をときどきそう入して、値を印刷させてみたりする方法を指導する。

最後に生徒の考えたいろいろな解法を検討して、効率よく実行できるプログラムは、どんなものかということまで話合えれば望ましい。

8. 基本外部関数と組込み関数

目 標

FORTRANでは基本外部関数(basic external function)や組込み関数(intrinsic function)が用意されているが、個々の関数の意味を理解して使用できるようにする。

内 容

次の7種類の基本外部関数を教える。

基本外部関数	定 義	引数の 個 数	英 字 名	引数の型	関数の型
指 数	e^a	1	EXP	実数型	実数型
自 然 対 数	$\log_e(a)$	1	ALOG	実数型	実数型
正 弦	$\sin(a)$	1	SIN	実数型	実数型
余 弦	$\cos(a)$	1	COS	実数型	実数型
双 曲 正 接	$\tanh(a)$	1	TANH	実数型	実数型
平 方 根	$(a)^{\frac{1}{2}}$	1	SQRT	実数型	実数型
逆 正 接	$\arctan(a)$	1	ATAN	実数型	実数型

次の6種類の組込み関数を教える。

組 込 み 関 数	定 義	引数の 個 数	英 字 名	引数の型	関数の型
絶 対 値	$ a $	1	ABS	実数型	実数型
		1	IABS	整数型	整数型
実 数 化	a	1	FLOAT	整数型	実数型
整 数 化	$\text{trunc}(a)$	1	IFIX	実数型	整数型
符 号 の 代 替	$ a_1 $ に a_2 の符号を付け たもの	2	SIGN	実数型	実数型
		2	ISIGN	整数型	整数型

指導上の留意点

組込み関数は、実際にやさしい数字を代入して演算してみせることにより意味を理解させる。これらの関数を含んだ算術代入文を多く練習させる。変数名、配列名、後で述べる関数名、サブルーチン名などには組込み関数や基本外部関数と同じ名前をつけてはいけないことを理解させる。

9. 演習問題(3)

目 標

基本外部関数と組み込み関数を利用したプログラムを書かせる。

内 容

(1) ヘロンの公式を使って三角形の面積を求める。

(2) 2次方程式の根を求める。

(3) これらの関数を含む適当な算術式の値を求める。

(1) 3辺を a, b, c とすると面積 S は、 $S = \sqrt{s(s-a)(s-b)(s-c)}$ で求められる
(ただしここで $s = \frac{a+b+c}{2}$ である)。データとして整数値を与えるとすれば、これを実数化するために `FLOAT` を用い、さらに `SQRT` を用いることになる。文を分けて書いてもよいが、引数自身がまた基本外部関数や組み込み関数であってもよいことに触れる。

(2) 2次方程式 $ax^2 + bx + c = 0$ の根は $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

この式をもとにプログラムを書かせる。

(3) ここでは、教師が適当に数式を与える。たとえば、次のようなものがある。

$$Z = \frac{1}{2} \log_e (x^3 + 1)$$

$$A = Be^{\sqrt{\frac{x}{r}}}$$

$$\theta = \cot \gamma - \sec \beta + \tan \alpha$$

$$C = |x| + \frac{|y|}{2} - \arctan(y)$$

指導上の留意点

生徒が演習問題に慣れてくるところと思われるので、データ・カードの設計だけでなく印刷の書式も生徒自身に自由に決めさせよう。レポート設計用紙を与え、見やすく、きれいな形で印刷できるように設計させて、それに合わせた `WRITE` 文を書かせる(レポート設計用紙については付録を参照)。印刷用紙の上段には、見出しや、作業日や、ページなどを印刷し、何段か印刷さ

せたらページ替えをすることなども考えさせる。データの個数を多くして、実際にそのルーチンを通過させ、うまく処理できたかどうかを検討させる。プログラムのデバグのときには、テスト・データをどう作るかが重要である。テスト・データには、そのプログラムのすべてのルーチンを通るような、あらゆる種類のデータを含めるようにする。

この辺で、流れ図を完全に書けるようにさせる。複雑なプログラムを作るときには、流れ図の必要なことを認識させ、以後、問題が与えられたら、まず流れ図を書く習慣をつけさせる。

またこれ以後、機械実習をするときには、入力データの書式の設計、印刷の書式の設計を生徒にやらせるようにする。

10. 文 関 数

目 標

文関数 (statement function) とはどんなものか、どのようなときに使用すると便利かを習得させる。

内 容

文関数定義文 (statement function defining statement) について、次のことを習得させる。

- (1) 文関数を定義する場所
- (2) 文関数定義文の形

実数型と整数型の区別、定義文に用いる式で許されるもの。

文関数の引用では、2つの引数、実引数 (actual argument) と仮引数 (dummy argument) について、個数、型、順序がどう対応しているかを理解させる。

例として $TAN(X)$ などをあげれば理解しやすいと思われる。

11. 演習問題(4)

目 標

文関数を使用して演習問題を行なう。

内 容

次のような問題が考えられる。

(1) $F(x) = \frac{8x+6}{3x+4}$ を定義し、 x を 0 から 10 まで 0.5 きざみで動かしながら、それぞれの値に対する $F(x)$ の値を求めて、関数表を作成する。

(2) 2変数関数の値を求める問題：たとえば、

$$f(x, y) = \sqrt{x^2 + y^2} \text{ を定義して、}$$

$$\alpha = \sqrt{a^2 + b^2}$$

$$\beta = \frac{1}{\sqrt{p^2 + q^2}}$$

$$r = 0.5 + \sqrt{(u+v)^2 + w^2} \text{ を求めること}$$

など。

指導上の留意点

今までの実習において、生徒からコンピュータを使って求めた数値と、実際に手で計算した結果とが異なるという質問を受けたかもしれない。そこでこの辺で、データがコンピュータ内部ではどのように表現されるかを理解させ、そのためにわずかながら数値の狂いが生ずる場合のあることを教える。

数表作成における印刷の形式は、F変換で指定するよりもE変換で指定する方が、印刷できる数値の範囲が大きくなり、便利であることを説明する。

E変換で印刷するときには、全体のけた数 w と小数点以下のけた数 d の指定は、 $w \geq d + 7$ の関係を満足するようにしなければならない理由を説明する。

12. 副 プ ロ グ ラ ム

目 標

主プログラムと副プログラムの関係、副プログラムを使用する利点を理解させる。副プログラムの役割とプログラム単位 (Program unit) の意味を説明する。主プログラムも副プログラムもそれぞれ1つのプログラム単位であることを理解させる。さらに副プログラムについて、関数副プログラム (function subprogram) とサブルーチン副プログラム (subroutine subprogram) の意味と使い方を説明する。

内 容

12.1 関数副プログラム

関数副プログラムの形式を習得させ、使いこなせるようにする。関数副プログラムを定義する上での規則について触れるとき、EQUIVALENCE文、COMMON文ということばが出てくるが、その詳細な説明は後にして、ここではことばだけにとどめる。RETURN文の説明をする。

関数副プログラムの引用の仕方を習得させる。このとき、実引数と仮引数の対応関係、実引数として許されているもの、主プログラムと副プログラムは別のプログラム単位であり、同じ変数を使用しても記憶場所は異なることを理解させる。

1つの文から成る関数を定義するのは、文関数でも関数副プログラムでも可能であることを、簡単な例で示すとよい。

12.2 サブルーチン副プログラム

サブルーチン副プログラムの形式を習得させ、使いこなせるようにする。

サブルーチン副プログラムの定義の仕方を教える。関数副プログラム名は値をもつが、サブルーチン副プログラム名は値をもたないので、両者を混同することのないよう指導しなければならないし、後者は値をもたないということから当然実数型、定数型の区別はないことを説明する。

サブルーチン副プログラムに書くことのできる仮引数の種類を説明し、さらに仮引数と実引数との間の対応関係を説明する。

サブルーチン副プログラムの引用の仕方を習得させる。サブルーチン副プログラムはCALL文で呼出することを説明する。

関数副プログラムとサブルーチン副プログラムの異なる点を指摘し、主プログラムに戻す値が2個以上必要な場合には、サブルーチン副プログラムでしか表わせないことを説明する。

13. 演習問題(5)

目 標

関数副プログラムや、サブルーチン副プログラムを定義し、それを引用する主プログラムを作成できる能力を養う。

内 容

次の問題を適当に選択して実習を行なう。

$$(1) \quad A = \sum_{k=1}^n k = \frac{n(n+1)}{2}$$

$$B = \sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$$

を求める副プログラムと、それを引用する主プログラムを作成する。

(2) 2次方程式 $ax^2 + bx + c = 0$ の根を求める副プログラムと、それを引用する主プログラムを作成する。

(3) 銀行預金の元利合計の計算をする。

元金を P 、年利率を r 、元利合計を A とすると、 n 年後の元利合計 A は、 $A = P(1+r)^n$ である。 P と n を引数として渡して、元利合計 A を求める副プログラムと、それを引用する主プログラムを作成する。

指導上の留意点

主プログラムから副プログラムを呼出す回数を複数回とし、副プログラムを作成する利点を理解させる。

1 4. COMMON文とEQUIVALENCE文

目 標

COMMON文とEQUIVALENCE文を習得させる。

内 容

1 4.1 COMMON文

COMMON文について次のことを習得させる。

- (1) COMMON文の意味
- (2) COMMON文を書く場所、宣言の仕方
- (3) 配列を共通ブロック (common block) にとるとき宣言の仕方
- (4) 主プログラムにおけるCOMMON文と副プログラムにおけるCOMMON文に現れる変数の対応の仕方

データをカードから読んで四則演算をするという程度の簡単な例で説明すれば、COMMON文の習得は容易であろう。

変数を共通ブロックにとる場合と、引数として渡す場合のプログラム上の表現の仕方の違いや、記憶場所の確保の仕方の違いなどを理解させる。

主プログラムと副プログラムで受け渡す変数は、共通ブロックにとるか、または引数にするかのどちらかにしなければならないこと、1つのプログラム単位にCOMMON文が複数個あるときの変数の対応の仕方、プログラム単位ごとにCOMMON文に書かれている変数の数が異なるときに対応の仕方などに重点を置いて説明する。

1 4.2 EQUIVALENCE文

EQUIVALENCE文について次のことを習得させる。

- (1) EQUIVALENCE文の意味
- (2) EQUIVALENCE文を書く場所、宣言の仕方、宣言したときどのような変数の対応関係ができるか。

EQUIVALENCE文はCOMMON文と異なり、あくまでも1つのプログラム単位内だけに関するものであることを理解させる。

15. 演習問題 (6)

目 標

COMMON文を使用したプログラムが作成できるようにする。

内 容

$$(1) \quad {}_n C_i = \frac{n(n-1)\cdots(n-i+1)}{i!}$$

を求める関数副プログラムと、それを引用する主プログラムを作成する。 n を共通ブロックにとる。

(2) 大きさ100の1次元配列の初めのN個の和と平均とを求めるサブルーチン副プログラムと、それを引用する主プログラムを作成する。

データの受け渡しは共通ブロックにとる変数と引数とに分ける。

(3) $n \times n$ の行列

$$C = \begin{pmatrix} C_{11} & C_{12} & C_{13} & \cdots & C_{1n} \\ C_{21} & C_{22} & C_{23} & \cdots & C_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & C_{n3} & \cdots & C_{nn} \end{pmatrix} \quad \text{の対角要素の和}$$

$X = \sum_{k=1}^n C_{kk}$ を求めるサブルーチン副プログラムと、それを引用する主プログラムを作成す

る。 n を共通ブロックで受け渡し、 C を引数として受け渡すことにする。

(4) 次のような内容のデータがある。

客コード 2けた(カラム1, 2) 1~10の数字

品目コード 2けた(カラム3, 4) 1~10の数字

数 量 3けた以内(カラム5~7)

単 価 3けた以内(カラム8~10)

原 単 価 3けた以内(カラム11~13)

データの終りを示すために、最後のデータの次に1~2けた目に-1の値をもった最終データがついている。

このデータを読込んで、次のことを行なう。

- (i) 客別(10種類)に金額を集計分類して印刷する。
- (ii) 品目別(10種類)に数量、金額、もうけを集計して印刷する。
- (iii) 客別、品目別に金額を集計する。
- (iv) 客別に利益を計算し、利益順位に配列し、印刷する。
- (v) 品目別に集計し、各品目の構成比率を算出する。

指導上の留意点

(1)は、
$${}_nC_i = \frac{(n-i)+1}{1} \times \frac{(n-i)+2}{2} \times \cdots \times \frac{(n-i)+(i-1)}{i-1} \times \frac{(n-i)+i}{i}$$

と変形すれば処理しやすくなる。

(2)は、配列のデータを主プログラムと副プログラムの間で受け渡すことになるので、両方のプログラムで同じ大きさに宣言しなければならない。

(3)も(2)と同様の注意を要する。

(4)は初めての事務計算なので、データは教師が用意した方がよいであろう。

プログラムの形は、主プログラムが(i)から(v)までの副プログラムを呼出すという形にする。データの受け渡しは、なるべくCOMMON文にした方が使用する領域が少なくすむので、そのように指導する。また数量や単価に大きな数を入れると、計算結果がけたあふれをおこす可能性があるので注意を要する。

客コードと品目コードは1から10までの数字なので、プログラムは比較的作りやすいはずである。これが0から9までの数字であれば、プログラムはもっと面倒になる。FORTRANでは、配列の添字の下限は1であるから、コードを扱うような問題のときには、1から始めた方がよいことを理解させる。また、データの書式を設計するときには、プログラミングしやすいようにする考慮が必要なことを理解させる。

16. 書式なし入出力文

目 標

書式なし入出力文 (unformatted input/output statement) の用途を習得させる。磁気テープを例にあげて説明すれば理解しやすいであろう。

内 容

16.1 書式なしREAD文

書式なしREAD文 (unformatted input statement) については、最も簡単な例：READ(U)(X(I), I = 1, 10) の程度のもを説明すればよい。

16.2 書式なしWRITE文

書式なしWRITE文 (unformatted output statement) については、最も簡単な例：WRITE(3)(X(I), I = 1, 10) の程度のもを説明すればよい。

17. 補 助 入 出 力 文

目 標

補助入出力文 (auxiliary input/output statement) を理解させる。

内 容

3種類の補助入出力文

REWIND文

BACKSPACE文

ENDFILE文

を簡単に説明する。書式なし入出力文と同様，磁気テープを例にとれば理解しやすいであろう。

18. 演習問題(7)

目 標

FORTRANの文法の説明を一通り終了したところで、総合演習という意味で生徒に自由に問題を解かせる。

内 容

難易度が演習問題(6)程度の問題を生徒に考えさせる。教師が問題を提供する場合には、いくつかの問題をあげて、生徒に自由に選ばせる。次のような問題が考えられる。

(1) 試験の問題

数教科の総合得点、得点順に配列、各教科の平均点、標準偏差などを求める。

(2) 給与計算の問題

残業時間、支払い給与額、給与袋入れ作業のとき各お札の枚数、平均給与、給与ランク別人数などを求める。

(3) 掃出し法で n 元1次方程式の根を求める。ここでは掃出し法による解の求め方の説明をする必要がある。

指導上の留意点

- (1) グループ作業とし、1チームの人数は3人から4人程度が望ましい。
- (2) 取扱う問題は生徒自身が考えたものでもよいし、教師が与えてもよい。生徒が考えたときには、教師はそれを聞いて、助言する必要がある。
- (3) 副プログラムに分割し、その1つを1人が完全に責任をもって完成させる。
- (4) コンピュータ処理の5段階——すなわち、問題の定義、分析、プログラミング、解決、文書化の5段階を完全に踏ませ、作業の結果をレポートにして提出させる。

19. ま と め

目 標

この講習で学んだFORTRANは基本的なもので、もっと多くの機能を持つFORTRANが広く使われていることを理解させる。

内 容

- (1) 全体を振り返ってまとめる。全体的な質問を受ける。
- (2) JIS FORTRAN 水準5000, 水準7000があることを話す。
- (3) コンピュータの規模とFORTRANの水準の關係に触れる。
- (4) 一般的に広く使われているFORTRANについて触れる。

参考文献

1. JIS 電子計算機プログラミング用言語 FORTRAN (水準3000)

JIS C 6203

日本規格協会

昭42.5 370円

このカリキュラム編成上の基本文献である。

2. やさしいFORTRAN

(財) 日本情報処理開発センター

昭43.10 550円

JIS 水準5000を基準としているので練習問題は少々むずかしいが、説明の展開の仕方、図などが参考になる。

3. 上級情報処理技術研修ガイドブック コンピュータ総合(基礎編)

通商産業省編

(財) 日本情報処理開発センター

昭45.3 2,600円

(II) 各論 第1部第4章 標準言語入門が参考になる。

4. プログラム学習によるFORTRAN

日本電信電話公社編

電気通信協会

昭43.4 1,400円

プログラム学習方式になっており、生徒の自習に役立つ。

5. JISに準拠したFORTRAN基本コース

監修 大泉 充郎

著者代表 高橋 理

オーム社

昭43.9 600円

内容は水準5000を少し下まわる程度であるが、練習問題によいものが多い。

6. わかりやすいFORTRAN

海老沢 成享 著

鹿島出版社

昭45.3 2,300円

「やさしいFORTRAN」と内容はほとんど同じである。

7. プログラム学習方式によるコンピュータ講座 FORTRAN基礎編(JIS 3000)

日立コンピュータ事業部

旺文社

昭45.2 1,300円

JIS 水準3000の習得をめざしているプログラム学習方式のテキストである。

8. FACOM FORTRAN入門

富士通(株)

実費

FORTRANの文法の説明がわかりやすい。

9. NEAC — シリーズ 2200

FORTRAN 基本例題集

日本電気㈱ 実費

関数表の作成などの例が載っている。

10. IBM S/360 FORTRAN IV (PI)

日本アイ・ビー・エム(株)

実費

水準は高いが、練習問題は参考になる。

第2部 プログラミング (2)

COBOL

ここでは次の用語を教える。

エラー・メッセージ, 机上デバグ, けたあふれ, 項目, 書式(形式), ステートメント(命令), 制御用文字, 直接編成, 定数, テスト・データ, 表(テーブル), リテラル, ループ, カウンタ(計数器), 順編成。

教育の目標

目標はCOBOLの基本的な書き方を習得し, 与えられた問題をCOBOLで記述できるようにすることである。

そのために, 次のようなことが必要である。

- (1) CODASYLのCOBOL 1965年版(以下COBOL-65という)のうち, 基本的な部分を理解させる。
- (2) 適宜機械実習を行ない, COBOLでいくつかのプログラムを組ませてみる。

COBOL-65は, 多くのコンピュータで使われており, COBOL文法書の基準になっている。COBOL-65の基本的な部分の文法を習得できれば, 実務について特定のコンピュータのCOBOL文法を学ばなければならなくなったとき, それを独力で習得できるはずである。

内容のあらまし

COBOL-65以前のCOBOLでは, 文法が必修部分(Required COBOL)と選択部分(Elective COBOL)とに分れていたが, COBOL-65ではこの区別がなくなっている。

COBOL文法について教える範囲を決めるに当たっては, 以前の必修部分と選択部分という区別を参考にした。必修部分を中心にし, 選択部分のうち, よく使われているものを加えてある。

COBOL-65のうち, 除かれた機能は次のようなものである。

- (1) 報告書作成機能

報告書セクション全体

INITIATE

GENERATE

TERMINATE

(2) 分類機能

SD記述

SORT

RELEASE

RETURN

(3) 非同期処理機能

SA記述

DECLARATIVES

PROCESS

HOLD

(4) 大記憶の処理

環境ディビジョン (Environment Division) のACCESS MODE句,

PROCESSING MODE句, ACTUAL KEY句など

手続きディビジョン (Procedure Division) のSEEK, READ, WRITE
のINVALID KEY句など

(5) 領域の共有

SAME AREA句

(6) プログラムの連結

連絡セクション (Linkage Section) 全体

CALL

RETURN

CANCEL

COPY句

(7) 翻訳プログラムへの指示

ENTER

EXIT

(8) 特殊な表索引

OCCURS DEPENDING ON句

USAGE IS INDEX句

INDEXED BY句

SET

SEARCH

(9) ラベルの処理

USEなど

(10) 特殊名

SPECIAL-NAMES段落全体

(11) データ・ディビジョン(Data Division)のレベル番号66の記述

SIZE句

SIGN句

SYNCHRONIZED句

POINT句

RANGE句など

(12) データ・ディビジョンの定数セクション

(13) 手続きディビジョンの

ALTER

TRANSFORM

EXAMINEなど

COBOL-65のうち、以上の機能を除いた部分を次のように分類して教える。

内 容	標準講義時間 (単位時間)	機 械 実 習 (標 準)	機 械 実 習 (最 低)
COBOL概論	10		
COBOLの歴史	1		
COBOLの特徴	1		
COBOLの言語体系	3		
表記法	1		
COBOLプログラムの構成	2		
COBOLプログラムの形式	2		
見出しディビジョンと環境ディビジョン	7		
見出しディビジョンの書き方	2		
環境ディビジョンの書き方	5		
演習問題(1)		10	2

内 容	標準講義時間 (単位時間)	機 械 実 習 (標 準)	機 械 実 習 (最 低)
データ・ディビジョン	28		
ファイルの構成	5		
データ・ディビジョンの書き方			
データ・ディビジョンの構成	3		
ファイル・セクションの書き方	15		
作業域セクションの書き方	5		
演習問題(2)		10	3
手続きディビジョン	55		
流れ図の復習	3		
手続きディビジョンの構成	2	5	
文の種類と命令の種類	2		
入出力命令	7		
移送命令	6		
演算命令	8		
処理順序変更命令	8		
表の処理	10	5	
演習問題(3)		40	
条件命令	8		
注釈命令	0.5		
翻訳指示命令	0.5		
演習問題(4)		50	
合 計	100	120	5

指導上の留意点

COBOLプログラムでは、

見出しディビジョン

環境ディビジョン

データ・ディビジョン

手続きディビジョン

の4つのディビジョンを書かなければならないので、4つのディビジョンを学んでからでなければコンピュータによる実習が十分に行なえない。しかし、それでは実習編の展開がむずかしいので、最初に簡単なサンプル・プログラムを提示し、4つのディビジョンについておおまかに教えておくとよい。その際手続きディビジョンについては、入出力命令・移送命令・演算命令についてのあらましを説明する。そうすれば、講義の間に適宜機械実習を折込むことができるであろう。

機械実習のための演習問題は、次のような時点で与えることにしたい。

見出しディビジョンと環境ディビジョンの説明が終ったとき	演 習 問 題 (1)
データ・ディビジョンの説明が終ったとき	演 習 問 題 (2)
手続きディビジョンの入出力命令・移送命令・演算命令・処理順序変更命令の説明が終ったとき	演 習 問 題 (3)
手続きディビジョンの説明が終ったとき	演 習 問 題 (4)

演習問題(1)では、データ・ディビジョンと手続きディビジョンをサンプル・プログラムに類似した簡単なものにしておき、生徒がプログラム全体を書けるように工夫する。

演習問題(2)では、データ・ディビジョンを複雑にし、手続きディビジョンは簡単なものを与える。

演習問題(1)と(2)では、4つのディビジョンの相互関係をよく理解させることが重要である。

演習問題(3)と(4)とは、4つのディビジョンを教えたあとで行なうことになるので、プログラム全体を独力で書かせることができるはずである。演習を通して、問題の分析やプログラム設計の能力も養い、プログラミング技術を身につけさせたい。

演習問題を行なうとき、グループを編成して共同で作業をさせることも考えられるが、その場合、1チームの人数は多くても4人程度にすべきである。グループ内では、相互に依存しあわないように、できるだけ作業を分担させ、分担した範囲の作業は、独力で行なうように指導すべきである。

COBOLについては、まだJIS規格が制定されていない。(JIS案は1971年3月完成——付録2-IV参照)。そのためにコンピュータによってCOBOLの文法や、使用できる機能はかなり異なるので、教師は機械実習に先だって、使用するコンピュータのCOBOL解説書を読み、機械実習の際に適切な助言が与えられるように準備しておかなければならない。

演習問題(1)、(2)で使用するテスト・データは、教師が用意する。演習問題(3)、(4)で使用するテスト・データは、生徒に作らせるのがよい。テスト・データは、プログラムのすべてのルーチンを通るように条件を考えて作らせる。

ジョブ制御カードが必要な場合には、教師が準備して生徒に渡し、その使い方を指示する。ジョブ制御カードの役割を簡単に説明するのはよいが、ジョブ制御カードの作り方については教える必要はない。

コンピュータを使用する場合、教師が生徒のエラーを発見したときの対し方には、慎重な配慮が必要である。コーディングしたものを最初にコンピュータに通すときには、コンピュータを使用する時間によほどの制約があるのでない限り、教師は生徒のエラーを指摘すべきではない。エラーを生徒自身の力で発見させることが大切なのである。生徒のエラー発見能力を見ながら、必要に応じて最小限度の助言を与え、デバッグ能力をのばすようにすることが望ましい。

デバッグの仕方も実習時に指導しなければならないが、これは演習問題(2)のときからにしたい。**机上デバッグ**を重視する考え方を身につけさせることが望ましい。相互参照テーブルについては、触れなくてもよいであろう。

1 COBOL 概論

目 標

COBOLの歴史、特徴などを説明し、COBOLの意義や役割を理解させる。

COBOL言語の仕組みやプログラムの形式を説明して、COBOLの輪郭を理解させ、COBOLに親しませる。

基礎編第3部第3章言語のあらましの復習を兼ねる。

内 容

1.1 COBOLの歴史

次のことを説明する。

- (1) COBOL言語を制定するにいたった動機や背景
- (2) 1959年にCODASYL(The Conference on Data System Languages)が設立されたこと、CODASYLがCOBOLの標準化と保全に果している役割
- (3) COBOLが拡張されて、

COBOL-60

COBOL-61

拡張COBOL-61

COBOL-65

などが発表された経過

1.2 COBOLの特徴

COBOLの特徴として次のことを説明する。

- (1) 共通言語(Common Language)であること。
これはアセンブラ言語と対比して例示すれば、わかりやすいであろう。
- (2) 事務用言語(Business Oriented Language)であること。
これはFORTRANなどの技術用言語と対比して例示すれば、わかりやすいであろう。
- (3) 文書化の便宜 — COBOL言語で記述されたプログラムは、ドキュメント(文書)として

も活用できることがCOBOL制定のねらいの1つであった。

(4) 英語との類似——英語のできる人にはわかりやすい。

1.3 COBOLの言語体系

COBOLの言語体系について説明する。次の2つのレベルに分けて説明するとよい。

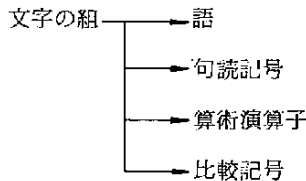
(a) 文字の組

(b) 語

(1) 文字の組

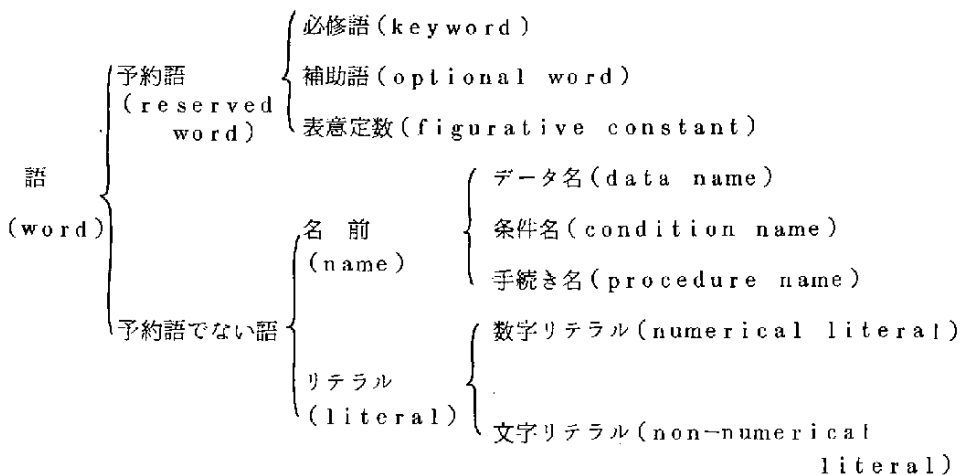
COBOLで使用される文字の組は、数字10字、英字26字、特殊文字15字から成ることを説明する。特殊文字については、その用例を句読記号、算術演算子、比較記号に分けて説明する。

文字の組と語とをはっきり区別して理解させることが必要である。次のような表を示すとよいであろう。



(2) 語

最初に語の種類を体系的に示す。そのため次のような表を書いて説明するとよい。



この表でいう名前とは、プログラマが付ける名前のことである。名前には、データ名、条件名、手続き名、特殊名 (special name) などがあるが、特殊名には言及する必要はない。

語の説明では、特に次のことに注意する必要がある。

- (a) 予約語では、必修語と補助語との違いを理解させる。
- (b) 名前では、一般的な命名規則を明らかにする。データ名、条件名、手続き名の違いを例をあげて説明する。
- (c) リテラルは、数字リテラルと文字リテラルに分れるが、数字と文字との区別は、COBOLでは特に重要であることを理解させる。できるだけ多くの例をあげて使い方を理解させたい。
- (d) リテラルと表意定数との区別を理解させ、表意定数の使い方を説明する。

語についての理解度を確認するために、次のような種類の質問をしてみるとよい。

- (a) 1765は、どんな種類の名前か。
- (b) 次のような名前の書き方は正しいか誤りか。

```
EMPL NO  
-AB5-  
CODE  
WORKING-SECTION  
Z
```

- (c) 次の文で使われているリテラルの種類を答えさせる。

```
ADD 1 TO ACCUMULATOR.  
MOVE '31416' TO GOKEI.
```

- (d) 次の文を実行したときのAREA-1の内容は何か。

```
MOVE SPACE TO AREA-1.  
MOVE 'A2' TO AREA-1.
```

1.4 表 記 法

COBOL言語の文法書や説明書で、一般的に使われている次のような表記の仕方を説明する。

- (1) 英字の大文字——予約語

必修語にはアンダーラインがあるが、補助語にはない。

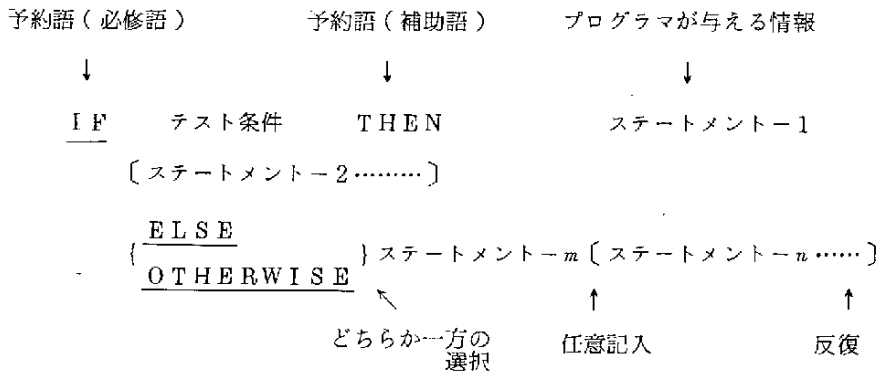
- (2) 日本語(または英字の小文字)——名前または命令のオペランド。

- (3) 大かっこ〔 〕——任意記入。〔 〕内は省略してよい場合もある。

(4) 中かっこ { } — 選択。 { } 内のどれか1つを選んで記入する。

(5) 点線…………… — 点線の前に書かれたことがらを繰返して記入する。

この初級カリキュラムでも必要に応じて上記の表記法を使用している。次のように例示するとよい。



1.5 COBOLプログラムの構成

COBOLプログラムを構成する要素について、次のように大きな単位から小さな単位へ順を追って説明する。

ディビジョン (division)

↓

セクション (section)

↓

段落 (パラグラフ) (paragraph)

↓

記述項 (entry)

4つのディビジョンを示し、その役割について簡単に紹介する。セクションと段落についての理解を深めるために、次のような角度から生徒に整理させてみるとよい。

セクションと段落を必ず含むディビジョン

セクションを必ず含むが、段落を含まないディビジョン

セクションと段落を含んでも含まなくてもよいディビジョン

セクションを含まないが、段落を含むディビジョン

1.6 COBOLプログラムの形式

簡単なサンプル・プログラムを示しながら、次のことを理解させる。

- (1) コーディング・シートは、ページ欄、行番号欄、続き表示欄、プログラム記入欄、標識欄に分れていること（COBOLコーディング・シートについては付録を参照）。
- (2) 任意記入（書いても書かなくてもよい）欄と、そうでない欄とがあること。
- (3) 続き表示が必要な場合の記入の仕方（これは機種によって異なる）。
- (4) プログラム記入欄には、A マージンとB マージンとがあり、A マージンで始る記述項とB マージンで始る記述項があること。

指導上の留意点

COBOLの歴史と特徴について、あまり深入りして教える必要はない。

COBOLプログラムの構成と形式については、サンプル・プログラムを示して説明するとよい。サンプル・プログラムとしては、カード・ファイルを読み、簡単な計算をして印刷する程度のものでよい。

2 見出しディビジョンと環境ディビジョン

目 標

- (1) 見出しディビジョンと環境ディビジョンの書き方を理解させる。
- (2) 問題の記述やシステム流れ図を見て、見出しディビジョンと環境ディビジョンが書けるようになる。

内 容

2.1 見出しディビジョンの書き方

見出しディビジョンの役割と書き方を説明する。

2.2 環境ディビジョンの書き方

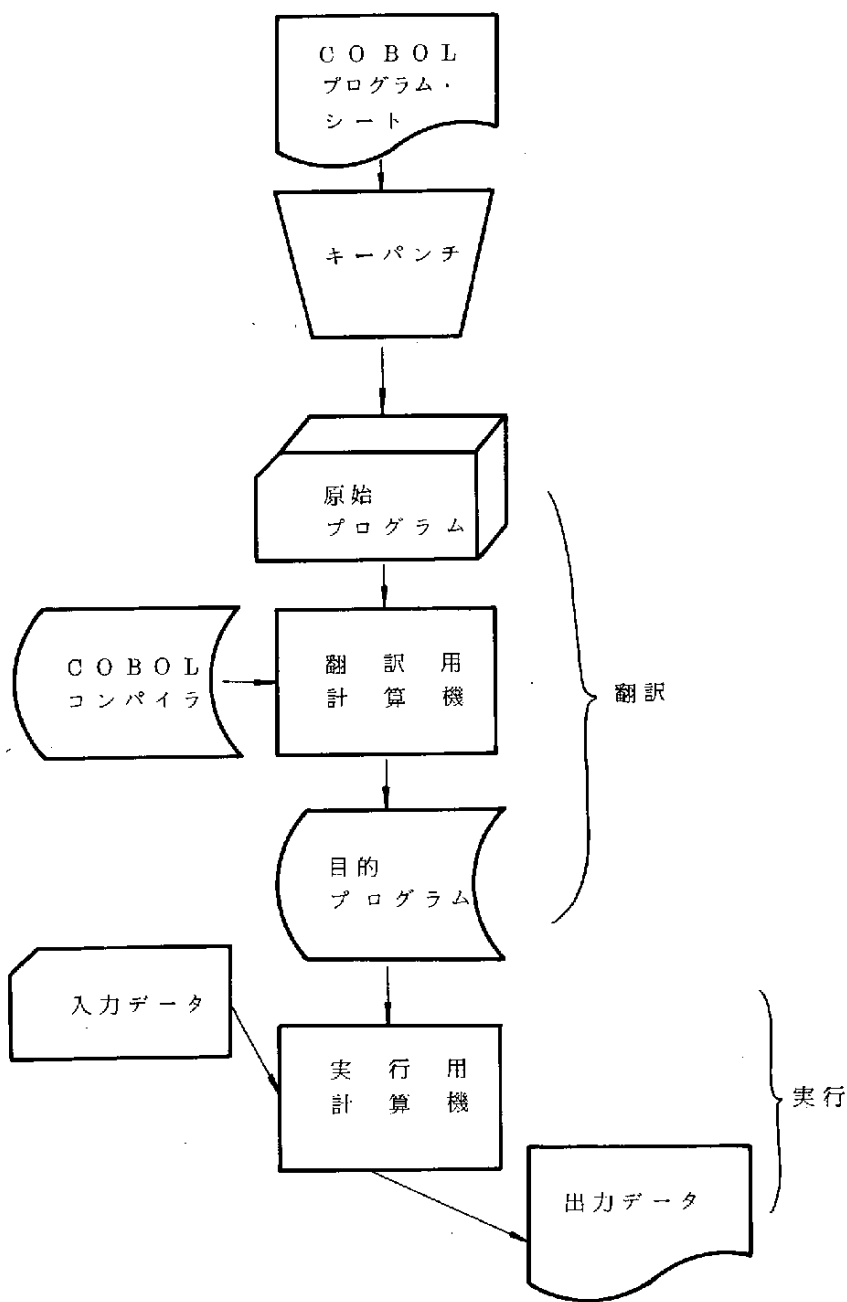
環境ディビジョンの役割、環境ディビジョンを構成する構成セクションと入出力セクションについて説明する。

2.2.1 構成セクション

構成セクション (CONFIGURATION SECTION) について、次のことがらを説明する。

- (1) 実行までの処理手順と構成セクションの役割
- (2) 構成セクションの書き方

次のようなシステム流れ図を示して、COBOLで書かれたプログラムで実行を行なうまでの手順と、翻訳用コンピュータ (ソース・コンピュータ) , 実行用コンピュータ (オブジェクト・コンピュータ) の役割を説明し、構成セクションの役割を理解させるとよい。



2.2.2 入出力セッション

入出力セッション (INPUT-OUTPUT SECTION) について、次のことがらを理解させる。

- (1) ファイルと入出力装置との結びつけ方
- (2) 入出力セッションの書き方

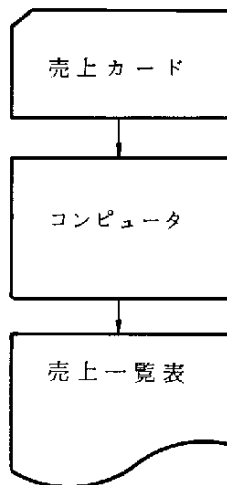
主としてFILE-CONTROL段落について説明する。

I-O-CONTROL段落は、ハードウェアによって役割が異なるので、実習に使用するコンピュータとの関係で説明の仕方を決めるとよい。

2.3 演習問題(1)

カードを読んで印刷するという程度の簡単な演習問題を与える。たとえば、次のような条件を与えてプログラムを書かせる。

システム流れ図



レコード形式

売上カード

商品 番号	得意 先 番号	セ ー ル ス マ ン 番 号	数 量	割 引 率	単 価	そ の 他
(5)	(8)	(3)	(5)	(3)	(5)	(51)

売上一覧表

商品番号	数 量	単 価	金 額	割引額	売 上 げ
(5)	(5)	(5)	(10)	(10)	(10)

演算処理

金額＝単価×数量

割引額＝金額×割引率× $\frac{1}{1000}$

売上げ＝金額－割引額

見出しディビジョンと環境ディビジョンは独力で書かせる。データ・ディビジョンと手続きディビジョンについては、サンプル・プログラムに対する説明の程度ともならみ合わせて、適当にヒントを与えるとよい。

指導上の留意点

環境ディビジョンは、ハードウェアに依存している部分が多いので、コンピュータによって書き方が異なる。したがって、コンピュータごとにどの程度異っているかを例示して説明し、実際にコンピュータを使う際には、そのコンピュータのCOBOL解説書を読む必要があることを指摘しておく。

3 データ・ディビジョン

目 標

- (1) データ・ファイルの構成について理解させる。
- (2) データ・ディビジョンの構成と文法を理解させる。
- (3) ファイル設計書を見て、データ・ディビジョンが書けるようにする。

内 容

3.1 ファイルの構成

データ・ファイルの種類とその論理的な構成について説明する。ファイルについての理解が不足しているとデータ・ディビジョンを理解することが困難なので、十分な説明が必要である。

(1) ファイルの種類

ファイルをいろいろな角度から区分して、例をあげながら説明し、ファイルという考えに慣れさせる。

たとえば次のような種類分けをするとよい。

(a) 記録媒体による違い

カード・ファイル
プリント・ファイル
テープ・ファイル
ディスク・ファイルなど

(b) 使用目的による違い

在庫ファイル
給与ファイル
顧客ファイルなど

(c) 機能による違い

基本ファイル
取引ファイルなど

(d) 編成方法による違い

順編成ファイル

直接編成ファイルなど

直接編成ファイルについては、順編成ファイルとの関係であらましを説明すればよい。索引順編成ファイルや区分編成ファイルについては触れる必要はない。

(2) ファイルの論理的構成

ファイルはレコードから成り、レコードは項目から成ることを説明し、ファイルの論理的構成について理解させる。カード・ファイルを例にとりて、ファイルとレコードと項目という3つの単位について説明すると、わかりやすいだろう。

(3) レコード

レコードの種類をあげ、それぞれの意味と使われ方を説明する。次のようなレコードについて説明する必要がある。

(a) ラベル・レコードとデータ・レコード

(b) 固定長レコードと可変長レコード

(c) 論理レコードと物理レコード

ここでは磁気テープを例にとりて、磁気テープ上のラベル・レコードの位置や役割、データ・レコードのブロック化などを説明する。例示をしながら、レコードがCOBOLプログラムの中で、どういう意味を持っているかについての理解を深めさせるようにしたい。

(4) 項 目

グループ項目・基本項目・独立項目などについて、それぞれの意味と役割を説明する。

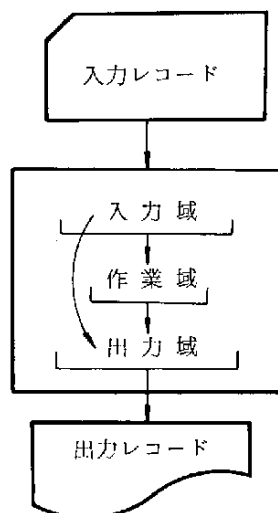
(5) 記憶域

記憶域について、次のことがらを説明する。

(a) 入力域と出力域

(b) 作業域

右のような図を示してデータ処理順序を説明し、入力域・出力域・作業域の役割を理解させる。



3.2 データ・ディビジョンの書き方

3.2.1 データ・ディビジョンの構成

- (1) データ・ディビジョンが、次のようなセクションから構成されることを説明する。

ファイル・セクション (FILE SECTION)

作業域セクション (WORKING-STORAGE SECTION)

定数セクション (CONSTANT SECTION)

連絡セクション (LINKAGE SECTION)

報告書セクション (REPORT SECTION)

- (2) ファイル・セクションと作業域セクションの役割について説明する。

定数セクション、連絡セクション、報告書セクションは、初級カリキュラムでは扱わないことにしているので、生徒が独習を希望するときは、参考資料を示すとよい。

3.2.2 ファイル・セクションの書き方

- (1) ファイル・セクションの構成

ファイル・セクションを構成する次の3つのレベルの記述について、それぞれの役割を説明する。

(a) ファイル記述

(b) レコード記述

(c) 項目記述

レコード記述と項目記述は、同じ形式である。COBOLではレコードと項目が同じ扱いになり得ることを説明して、(1)ファイル記述、(2)レコード記述と項目記述の2つに分けて理解させてもよい。

- (2) ファイル記述

次の記入項目の意味を理解させ、その書き方を習得させる。

(a) FDというレベル番号

(b) RECORDING MODE文節

(c) BLOCK CONTAINS文節

(d) RECORD CONTAINS文節

(e) LABEL RECORDS文節

(f) DATA RECORDS文節

レベル番号についての詳しい説明は、レコード記述のところで行なうことにして、ここではレ

ベル番号はFDについてだけ教えればよい。

(3) レコード記述と項目記述

レコード記述と項目記述は、文法上ではレベル番号で区別されることを説明する。

次の記述項目の意味を理解させ、その書き方を習得させる。

- (a) レベル番号
- (b) データ名
- (c) PICTURE文節
- (d) USAGE文節
- (e) VALUE文節
- (f) REDEFINES文節
- (g) OCCURS文節

レベル番号の次は原則としてデータ名を書くが、データ名の代りにFILLERまたは条件名を書くことがあることも説明しておかなければならない。

PICTURE文節は、内容が広いので、ここではそのすべてを説明しても理解が困難であると思われる。次のことがらに重点をおいて説明するのがよい。

PICTURE文節の役割

項目の種類とPICTURE指定の関係

PICTURE指定で使われる記号の意味。主として英数字項目、英字項目、数字項目の書式(形式)をPICTURE指定で定義するときに必要な記号について説明する。報告書項目は、手続きディビジョンでMOVE命令を説明するときにあわせて行なうのがよいであろう。

レベル番号の使い方については、よく理解させる必要がある。問題を与えて理解度を確かめ、さらに生徒が独力で記述できるようになるまで指導すべきである。

少なくとも次のような構造のレコードについて、それぞれの項目を独力で記述できなければならない。

例 1

販 売 レ コ ー ド	科 目	種 別	
		番 号	
	コ ー ド		
	明 細	得 意 先	住 所
		日 付	名 前
			年
月			
		日	
	数 量		
	単 価		

例 2

F 1	F 2	F 3	F 4	F 5	項 目 1
					項 目 2
					項 目 3
					項 目 4
					項 目 5
					項 目 6

さらに次のような例題をいくつか与えて、PICTURE文節の意味と書き方を習得させる必要がある。

例1.

項目記述の例を示し、項目の種類とけた数を答えさせる。

03 SHIPPING PICTURE XXX.

例2.

ある項目に1287というデータが貯えられているとして、この項目が次のようなPICTURE指定を持つとき、データがどのような数値として扱われるかを答えさせる。

- 1) 9(4)
- 2) VP(4)9(4)
- 3) 999V9
- 4) 9(3)P(4)V

USAGE文節は、コンピュータによって書き方が異なるので注意を要する。特に次のことについて、理解を確かめる必要がある。

10進数と2進数の表わし方

USAGE指定を省略したときの解釈

PICTURE指定とUSAGE指定との関係

VALUE文節では、初期値や条件値の与え方を理解させる。

ファイル・セクションで使うときは制限があるので、詳しいことは作業域セクションで説明する。

REDEFINES文節では、再定義の仕方を例をあげて説明する。

OCCURS文節では、表(または配列)のとり方を説明する。詳しいことは、手続きディビジョンの表(テーブル)の処理のところで説明する。

3.2.3 作業域セクションの書き方

ここでは、次のことがらを説明する。

(1) どのような場合に作業域セクションが必要か、作業域、定数、カウンタ(計数器)などの定義の仕方を説明する。

(2) 作業域セクションの書き方

独立項目と他の項目との記入順序についても説明する。

(3) ファイル・セクションと作業域セクションの関連

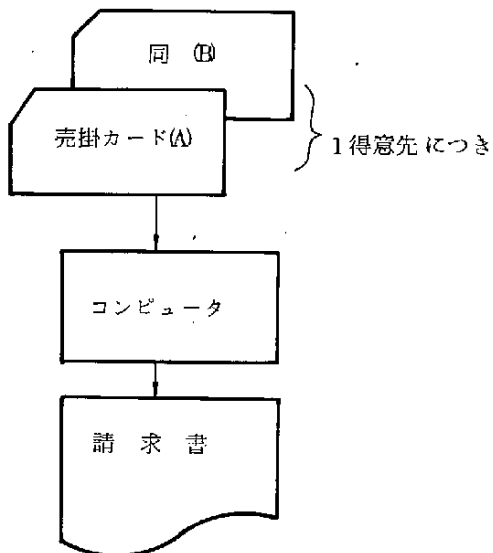
VALUE文節は、作業域セクションで広く使われるので、復習をし、例をあげて使い方を説明するとよい。

3.3 演習問題 (2)

データ・ディビジョンの書き方について、理解を深めるために演習問題を与える。

次のような問題が考えられる。

システム流れ図



レコード形式

売掛カード(A)

コード A	取引 種別	店番号	得意先 番号	得意先		ブ ラ ン ク
				得意先名	住所	

英
(1)

英数
(1)

数
(3)

数
(4)

英 数
(22)

英 数
(44)

英数
(5)

売掛カード(B)

コード B	取引明細										プランク	
	店番 号	得意先番 号	前月		当月							
			取引 数	売掛 残	請求 日	売掛		支払		返品		
						件 数	金 額	件 数	金 額	件 数		金 額

英 数 数 数 数 数 数 数 数 数 数 数 英 数
(1) (3) (4) (2) (6) (6) (2) (6) (2) (6) (2) (6) (34)

請 求 書

得意先名	住 所	会 計 番 号
× ——— ×	× ————— ×	××× ××××
前月売掛残	売 掛	支 払
× ——— ×	× ——— ×	× ——— ×
		返 品
		× ——— ×
		請 求 額
		× ——— ×

演算処理

$$\text{請求額} = \text{前月売掛残} + \text{売掛} - \text{支払} - \text{返品}$$

データ・ディビジョンの書き方では、1 ファイル中に2種のレコードがある場合の記述の仕方
や、項目の従属関係（または包含関係）が正しく表現できているかどうかには注意を払う必要があ
る。

手続きディビジョンについては、生徒の力を見て適宜ヒントを与えるようにする。

指導上の留意点

データ・ディビジョンは、例をあげて説明をしないと理解させることが困難なので、実務の中から具体的な例を選び、その例に沿って説明するのが望ましい。

4 手続きディビジョン

目 標

- (1) 手続きディビジョンの構文と文法を理解させる。
- (2) 問題記述書を見て、プログラム流れ図を書き、さらにそれをもとに手続きディビジョンが書けるようにする。

内 容

4.1 流れ図の復習

基礎編第4部第1章で学んだ流れ図について復習し、流れ図について演習問題を与える。

次のような内容を含む流れ図をテンプレートを使って書かせてみるとよいであろう（流れ図記入用紙とテンプレートについては付録参照）。

- (1) 明細印刷
- (2) 合計印刷
- (3) 突合せ
- (4) スイッチング

システム流れ図とファイル設計書を見て、プログラム流れ図を書けるようにすることが大切である（ファイル設計用紙については付録参照）。

事務計算のプログラミングでは、プログラム言語についての理解と同時に、流れ図を書く能力が重要である。

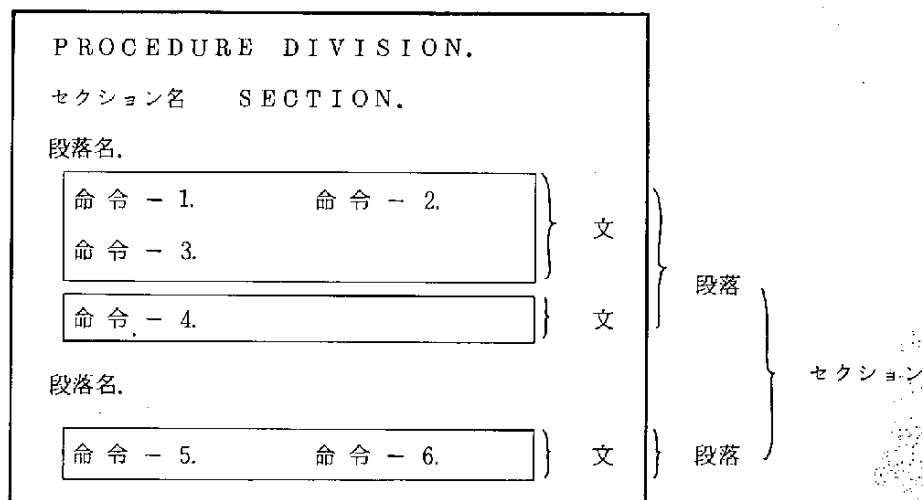
流れ図を書く能力が不十分なまま手続きディビジョンに進むことになると、教育効果はあがらない。

したがって、教師は演習問題を解かせながら、生徒の力をよく見きわめて先に進むようにしなければならない。

4.2 手続きディビジョンの構成

セクション、段落(paragraph)、文(sentence)、ステートメント(命令)(statement)の特徴を説明する。

図を示して、セクションが段落から、段落が文から、文が命令から構成されることを理解させる。たとえば、次のような図を示す。



生徒の理解度を知るためにサンプル・プログラムを用意し、特に手続きディビジョンの部分を示して、命令、文、段落などがそれぞれの部分になるかを答えさせるとよい。

4.3 文の種類と命令の種類

- (1) 文の種類を条件文と非条件文に分けて説明する。
- (2) 命令の種類を整理して説明する。

たとえば、次のような整理の仕方が考えられる。

無条件実行命令

入出力命令

移送命令

演算命令

処理順序変更命令

条件命令

注釈命令

翻訳指示命令

4.4 入出力命令

入出力命令については、レコードとして処理する場合と、単一の独立したデータとして処理する場合とに分けて説明する。

(1) レコードとして処理する場合

レコードの入出力を含んだ流れ図を示し、それに従ってプログラムを書くとき、次のような入出力命令をどう書くかを説明する。

OPEN命令

CLOSE命令

READ命令

WRITE命令

(2) 単一の独立したデータとして処理する場合

単一の独立したデータとして処理する場合の流れ図を示し、次の入出力命令の書き方と使い方を説明する。

ACCEPT命令

DISPLAY命令

READ, WRITE命令を使って入出力処理を記述するときと、ACCEPT, DISPLAY命令を使って入出力処理を記述する場合とでは、プログラムの書き方が非常に違っている。

	READ, WRITE による記述	ACCEPT, DISPLAY による記述
環境ディビジョンのSELECTの指定	必要	不要
データ・ディビジョンのFDの指定	必要	不要
データを入出力する場所	入出力域	作業域
データの読取り書出し命令	READ, WRITE	ACCEPT, DISPLAY
OPEN, CLOSE	必要	不要

一般的にACCEPT, DISPLAYは、コンピュータと操作員との間の情報のやりとりのために使われ、通常の入出力処理には、READ, WRITE命令を使わなければならないことを理解させる。READ命令と関連させて、AT ENDを説明する。スペースとスキップの制

御について説明する。制御用文字に触れる。

4.5 移送命令

MOVE命令を説明する。

MOVE命令について、次のことを理解させる。

(1) データを移送するときの制約

英数字項目は、数字項目へ移送できないことなど。

(2) 編集の仕方

編集の仕組みと報告書項目の記述の仕方、報告書項目へMOVEしたときの変化など。

(3) MOVE命令とMOVE CORRESPONDING命令の使い方

4.6 演算命令

演算命令は次の2組に分けて説明する。

(1) ADD命令

SUBTRACT命令

MULTIPLY命令

DIVIDE命令

(2) COMPUTE命令

(1)の各命令は、同じ形式をしているので、ADD命令を例にとって次のことがらを説明するとよい。

(a) 被加数と加数の位置

(b) 四捨五入の指示の仕方

(c) けたあふれの場合の処理の仕方

(d) 答を求める場所を変えることができること。

(2)のCOMPUTE命令については、次のことを理解させる。

(a) 算術演算子の種類

単項演算子を含める。

(b) 演算の優先順位

(c) かつこの使い方

4.7 処理順序変更命令

ここでは、GO TO命令、PERFORM命令、STOP命令を説明する。

GO TO命令については、分岐という概念を理解させる必要がある。例をあげて説明するとよい。

PERFORM命令に関連して、次のことを説明する。

- (1) サブルーチンの処理
- (2) PERFORMの書き方と使い方
- (3) PERFORM命令を使うときの制約サブルーチンの実行範囲の交差の問題 など。

PERFORM命令は、プログラミングについての基礎的な技術がなければ理解がむずかしい。サブルーチンや繰返しの処理(ループ)について多くの例や問題を示して、十分理解させる必要がある。

例題として次のようなものが考えられる。

- (1) STEP-1からSTEP-10までの段落を20回繰返す。
- (2) 100から1,000までのすべての整数の和を求める。

STOP命令については、一時停止の場合と、完全な停止の場合との命令の書き方を理解させる。

4.8 表の処理

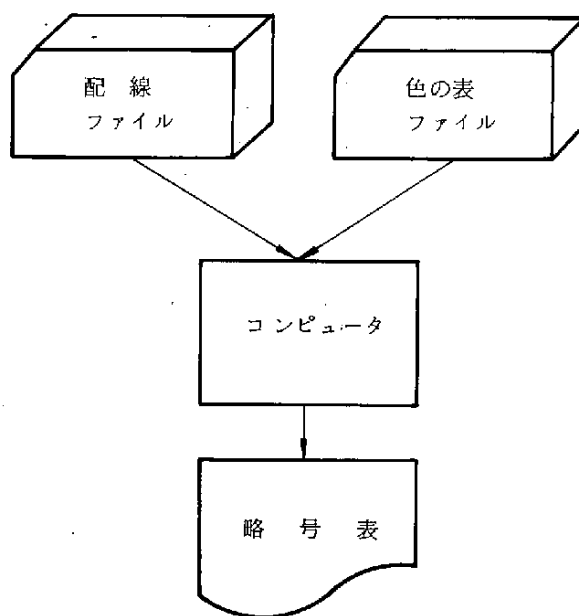
- (1) 表の定義
 - (a) 例をあげて、表の意味を説明する。
 - (b) 表の次元について説明する。
 - (c) 表の定義の仕方を説明する。OCCURS句を使って定義する方法について説明する。表と添字の関係を説明し、添字の使い方を理解させる。
- (2) 表引き
 - (a) 表引きの仕方を説明する(線型探索と二分探索)。
 - (b) COBOLプログラムでの表引きの仕方をIF文を使って説明する。
 - (c) IF文の代りに、PERFORM命令を使って表引きを行なう方法を説明する。

4.9 演習問題(3)

手続きディビジョンでは、PERFORMの使い方と表の処理についての理解が大切である。

ここで、たとえば次のような演習問題を与えて、プログラムを書かせるとよい。

システム流れ図



レコード形式

配線ファイル

配線番号	位置	色略号	その他
(5)けた	(3)	(1)	(71)

色の表ファイル

色略号	色	その他
B	BLACK	
W	WHITE	
R	BROWN	
G	GREEN	
L	BLUE	
(1)けた	(5)	(74)

略号表

配線番号	位 置	色
×××××	×××	×××××

表は、色の表ファイルを読んで作成するが、色の表ファイルを読むかわりにプログラム内で VALUE 文節を使って表を作成させてもよい。

1次元の表の処理を理解できた生徒には、各色ごとに長さが異なる3種の配線があると仮定して、2次元の表の処理を行なわせるとよい。

4.10 条件命令

IF文の例をあげて、条件命令の一般的な書き方を説明する。ELSEやOTHERWISEを含まない場合と含む場合とでは処理順序が異なること、文の切れ目、すなわちピリオドの位置が重要な意味を持つことを理解させる。

条件文を次のような種類に分けて、その使い方を説明する。

- (1) 比較テスト
- (2) 符号テスト
- (3) クラス・テスト
- (4) 条件名テスト
- (5) ページあふれテスト

比較テストでは、数字の比較と文字の比較との違いを説明する。

IFのテスト、条件の重複(AND, ORを用いる場合)については、触れなくてもよい。

条件名テストでは、データ・ディビジョンにおける条件名の定義と関連させて説明する。

ページあふれテストは、コンピュータによって扱い方が異なることがあるので注意する。

4.1 1 注釈命令

NOTE命令の書き方と使い方を説明する。

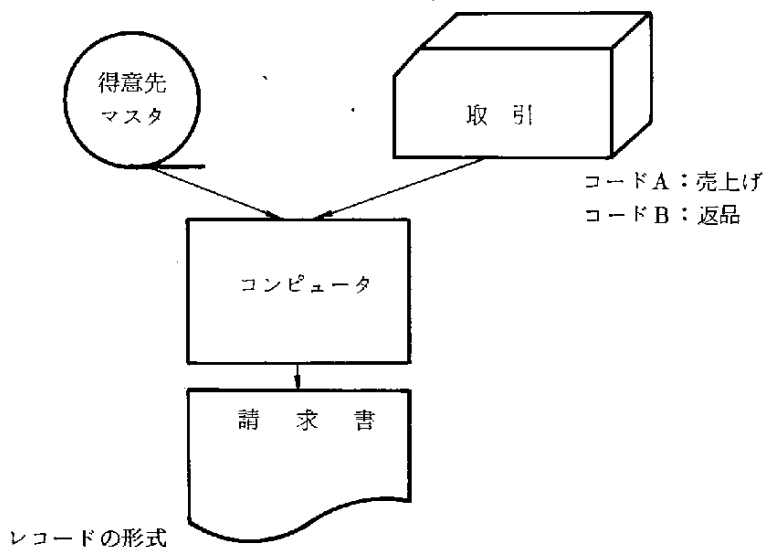
4.1 2 翻訳指示命令

翻訳指示命令の役割を説明し、COPY命令、INCLUDE命令について簡単に説明する。

4.1 3 演習問題(4)

次の問題と同程度の水準の問題を与えて、演習を行なう。

システム流れ図



レコードの形式

得意先マスタ

得意先番号	得意先名	売掛残高	その他
(5)けた	(20)	(10)	(65)

取引

コード	日付			得意先番号	品名	単価	数量	その他
	年	月	日					
(1)	(2)	(2)	(2)	(5)	(15)	(5)	(6)	(42)

1 けた目のレコードが A であれば売上げ、B であれば返品を表わす。

請 求 書

請 求 書				
PAGE. 1				
貴店名 OOKUBO SYOJI 殿				
日 付	品 名	単 価	数 量	金 額
46/4/1	ZAN	5,000	1	5,000
46/4/25	STAND	29,000	2-	58,000-
46/4/28	REIZOKO	62,500	3	187,500
	合 計			134,500

演算処理

売上げ（または返品）＝単価×数量

請求合計＝売掛残高＋売上げ金額－返品金額

上記の計算を行なって、請求書に印刷する。このためのプログラム流れ図を作成させ、それに従ってプログラムを書かせる。

指導上の留意点

手続きディビジョンを説明する際には、流れ図を書くこと、流れ図に従って手続きディビジョンを書くことを結びつけて教えることが重要である。

生徒の流れ図を書く力に相当差があるときには、演習のときにできるだけ個別指導をし、生徒の能力に応じて適切な助言を与えることが望ましい。

MOVE 命令のところで編集の仕方を教える際には、報告書項目の記述の仕方についてよく使われるものを重点的に教えるようにしたい。

演習問題(3)、(4)については、文書化が重要であることを徹底させて、よい文書を作らせるように指導したい。演習の成果は、文書にして提出させる。演習の中でデータ・エラーのチェックの仕方やエラー・メッセージの使い方なども指導する。

システム流れ図、ファイル、レコードの形式などを未定のまま問題を与え、それを生徒に設計させてもよい。レポートの形式を設計させるときには、レポート設計用紙を使うとよい(レポート設計用紙については付録 参照)。

参考文献

1 COBOL 1965年版

アメリカ国防省

関根智明 監訳

情報処理学会

昭43 3,000円

CODASYLのCOBOL委員会が作成した報告書を翻訳したもので、このカリキュラムの基礎資料である。

2 プログラム学習によるCOBOL I 入門編

日本電電公社編

電気通信協会

昭43 550円

プログラム学習によるCOBOL II 基礎編(1)

日本電電公社編

電気通信協会

昭43 650円

プログラム学習によるCOBOL III 基礎編(2)

日本電電公社編

電気通信協会

昭43 600円

プログラム学習によるCOBOL IV 基礎編(3)

日本電電公社編

電気通信協会

昭43 700円

プログラム学習によるCOBOL V 応用編

日本電電公社編

電気通信協会

昭43 650円

初心者向けに好適である。

3 やさしいCOBOL

(財) 日本情報処理開発センター 実費

昭和45 550円

4 COBOLの学び方

日本ソフトウェア機編

電機大学出版局

昭45 420円

高校生向けの教材として教科書的に書かれている。

5 プログラマーのためのCOBOL入門

マクラケン

小岩 明, 岡 祐 共訳

竹内書店

昭44 1,500円

手続きディビジョン, データ・ディビジョン, 環境ディビジョン, 見出しディビジョ
ンの順に説明している。説明の仕方が参考になる。

6 HITAC 8300/8400/8500 COBOLプログラム・マニュアル

機 日立製作所 実費

昭44 ユーザ向けマニュアル

7 FACOM COBOL 解説編

富士通機 実費

昭44 ユーザ向けマニュアル

8 NEACシリーズ2200 COBOL 説明書

日本電気機 実費

昭44 ユーザ向けマニュアル

9 システム/360オペレーティング・システム COBOL言語解説書

日本アイ・ビー・エム機 実費

昭43 ユーザ向けマニュアル

10 COBOLプログラムの基礎

COBOLプログラムの書き方

COBOLプログラムの手法

日本アイ・ビー・エム(株) 実費

プログラム学習方式による初心者向けの独習書である。

第 3 部 データ準備

ここでは次の用語を教える。

オーバパンチ、オフパンチ、カード・ゲージ、検孔、けん盤、視覚検査、集団せん孔、レースパンチ

教育の目標

コンピュータを使ってデータ処理をするときには、コンピュータに入力するデータをいかにして正確に、かつ、すみやかに作成するかということが重要になる。入力データを正確にするためには、データのチェックという作業も必要である。

現在のコンピュータでは、データをコンピュータに入力する媒体として、紙カードと紙テープが多く利用されている。したがってこの部では、紙カードや紙テープ上にデータを記録する方法、データの正誤のチェックの仕方、紙カード、紙テープの取扱い上の注意、保管方法などを、実習を通して理解させることを目標にする。

内容のあらまし

紙カードを中心にして、データの準備の仕方を教える。紙テープについては特に必要でなければ教えなくてもよい。紙テープの知識も必要であると思われるときには、適宜内容を選択して教える。指導時間数は次のとおりである。

内 容	時 間 数	内 容	時 間 数
紙カード		紙テープ	
紙カードの作成	15	紙テープの作成	必要に応じ て時間数を 決める。
紙カードのチェック	10	紙テープのチェック	
紙カードの取扱い上の注意	3	紙テープの取扱い上の注意	
紙カードの保管方法	2	紙テープの保管方法	

1 紙 カ ー ド

1.1 紙カードの作成

目 標

なまデータやコーディング用紙に記入されているプログラムを、けん盤せん孔機を使って正しく紙カードにせん孔する方法を習得させる。少量のプログラム・カードやデータ・カード(100枚程度)なら、独力でせん孔できる力を身に付けさせる。

内 容

けん盤せん孔機の操作に関連して、次の各項目の説明をする。

(1) けん盤せん孔機の各部の名称と機能

- (a) カード・ホッパ
- (b) プレッシュャ・プレート
- (c) カード・ベット
- (d) パンチング・ステーション
- (e) リーディング・ステーション
- (f) カード・スタッカ
- (g) プログラム・ユニット
- (h) プログラム・コントロール・レバー
- (i) カラム・インディケータ
- (j) プレッシュャ・ロール・リリース・レバー
- (k) バック・スペース・キー
- (l) デスク(リーディング・ボード)
- (m) メイン・ライン・スイッチ
- (n) チップ・ボックス

(2) けん盤(キーボード)

せん孔用のキーについて簡単に説明してから、操作用のキーについてそれぞれの機能を述べる。

- (a) ニューメリック・シフト・キー

- (b) アルファベティック・シフト・キー
- (c) デュプリケート・キー
- (d) リリース・キー
- (e) フィード・キー
- (f) レジスタ・キー
- (g) スキップ・キー
- (h) マルチプル・パンチ・キー
- (i) プログラム・ワン・キー
- (j) プログラム・ツー・キー
- (k) レフト・ゼロ・キー
- (l) エラー・リセット・キー
- (m) オグジリアリ・デュプリケート・キー
- (n) マスタ・カード・キー
- (o) スペース・バー

(3) ファンクション・コントロール・スイッチ

個々のファンクション・コントロール・スイッチについて簡単にその機能を説明する。

- (a) オート・フィード・スイッチ
- (b) オートマティック・スキップ・アンド・デュプリケート・スイッチ
- (c) プログラム・セレクション・スイッチ
- (d) プリント・スイッチ
- (e) レフト・ゼロ・プリント・スイッチ
- (f) クリア・スイッチ

(4) プログラム・ユニット

プログラム・ユニットの機能と使用の仕方を説明する。

- (a) プログラム・カード
- (b) プログラム・コード

次の基本的な命令を与えるために必要な操作法を説明する。

- (i) 欄の限定
- (ii) 自動スキップの開始
- (iii) 自動複写の開始

(Ⅳ) 英字せん孔への切替え

(Ⅴ) レフト・ゼロの自動せん孔

(c) カードをドラムに巻きつける方法

(d) ドラムを機械にセットする方法

(5) スキップの操作

スキップの意味を説明し、次の2つのスキップの方法を述べる。

(a) 自動スキップ

(b) 手動スキップ

(6) 複写の操作

複写の意味と次のそれぞれの複写の仕方について説明する。

(a) 自動複写

ここで、集団せん孔についても触れる。

(b) 手動複写

複写と関連して、誤せん孔カードの訂正の仕方を説明する。

(7) レース・パンチについて

(8) レフト・ゼロ自動せん孔の操作

レフト・ゼロ自動せん孔の機能と、操作の仕方を簡単に説明する。

(9) 操作の開始と停止について

(a) 操作の開始

紙カードをカード・ホッパにセットしてからせん孔作業に入るまでの前準備について必要なことを説明する。

(b) 操作の停止

操作を停止したあとの後始末の仕方を説明する。

(10) ジャム・カードの取除き方

ジャム・カードの意味を説明し、紙カードがホッパからうまくフィードしないときの処理、紙カードがパンチング・ステーションやリーディング・ステーションでジャムしたときの処理方法を述べる。

(11) カード・ゲージ(card gauge)の使用目的と使用の仕方を説明する。オフパンチについて説明する。多数の紙カードをせん孔するときには、カード・ゲージを使ってあらかじめせん孔位置が正しいかどうかの点検作業をしてから、紙カードのせん孔をする習慣をつけさせるよう

指導上の留意点

- [illegible]

- 205 —

故障したときには、専門のキー・パンチャや、担当の人に申出るように指導する。

(7) プログラム・ユニットの使用方法を教えるときには、あまり複雑なものは省いて、スキップ開始、複写の開始、欄の限定などを中心にして教えればよい。また、次の点に注意を払う必要があることを述べる。

(a) プログラム・ドラムをプログラム・ユニットから出し入れするときは、プログラム・コントロール・レバーをOFFにすること。

(b) プログラム・ドラムに紙カードが巻かれていないときには、必ずプログラム・コントロール・レバーをOFFにすること。

(c) プログラム・コントロール・レバーをONにしてせん孔を始めたら、せん孔途中でプログラム・コントロール・レバーをOFFにしたり、またONにしたりする切替えをしないこと。

(8) エラー・リセット・キーの使い方。

(9) 29コードと26コードの違いを説明する。

(10) かたかなせん孔にも簡単に触れる。

(11) キー・パンチの練習をさせるとき、第1部のFORTRAN編や第2部のCOBOL編で実際にプログラムやデータを記入したコーディング用紙を参照するとよい。

1.2 紙カードのチェック

目 標

紙カードにデータが正しくせん孔されたかどうかをチェックする方法を習得させる。

内 容

次のようなチェックの方法を説明する。

(1) 視覚検査(sight-check)

せん孔される前のデータと紙カードにせん孔されたデータとを比較する。間違っているときには、部分的に(正しい部分は複写する)あるいは全体的に打直す。打直した紙カードを、間違っている紙カードの上に重ねて正しく訂正できたかどうかを目で確かめてから間違いの紙カードを処分する。

(2) 検 孔(verify)

せん孔された紙カードを検孔機でチェックする。

(3) コンピュータでチェックする方法

(a) コンパイルまたはアセンブル・エラーのメッセージ

(b) ユーティリティ・プログラムを使用して、紙カードの内容を印刷させて目で検査する。

(c) プログラムの中でデータの誤りをチェックする。次のようなチェックが考えられる。

(i) データの書式(形式)のチェック

(ii) 順序のチェック

(iii) 検査数字によるチェック

(iv) リミット・チェック

データが許された範囲内にあるかどうかをチェックする。

(v) バリディティ・チェック

(vi) バッチ・トータル・チェック

レコード数や金額の合計などを調べてチェックする。

(vii) バランス・チェック

項目の加減算を行なってチェックする。

指導上の留意点

- (1) チェックの仕方を理解すると同時に、どうしたら間違いのないデータを作成することができるかを考えさせるようにする。間違いのないデータを作成することは、データ量が多量になり、複雑になるほど困難になるが、せん孔前のデータをわかりやすく、きれいな文字で、注意深く書くことによって、間違いをかなり避けることができる。
- (2) 誤せん孔カードの特定のけただけ打直す方法を実習させる。
- (3) 検孔機は、実習しなくてもよい。機械を見せて、検査される仕組みを説明するだけでよい。
- (4) プログラムでデータをチェックする方法を説明するのは簡単でよい。プログラムによるデータ・チェックは、紙カードの設計の仕方と深い関係があることを理解させる。

1.3 紙カードの取扱い上の注意

目 標

紙カードを取扱うときに注意しなければならないことを理解させる。

内 容

次のようなことを中心に述べる。

- (1) 紙カードを折ったり曲げたりしない。
- (2) 紙カードを閉じたり、他の紙カードや書類などと一緒にクリップで止めたりしない。
- (3) 多くの枚数の紙カードを持運ぶときは、箱に入れるなどして、じかに手で持たない。
- (4) 1枚目の紙カードやカード・デックの上面にデータの内容を示す簡単な表示を入れる。
- (5) 紙カードの静電気を取除くために、紙カードの左右をよくさばく。

指導上の留意点

- (1) 単に禁止事項を羅列するだけではなく、なぜそのようなことに注意を払う必要があるかを説明して理解させる。
- (2) 実際に紙カードを用いて実演してみせる。

1.4 紙カードの保管方法

目 標

正しい紙カードの保管方法を習得させる。

内 容

次の各項目について説明する。

- (1) 温度、湿度、その他保管室の環境について
- (2) カード・キャビネットでの保管方法
- (3) 正しくせん孔された紙カード、未使用の紙カードの区別、紙カードの整理整頓

指導上の留意点

- (1) 湿気が多い場所に保存すると、紙カードが水分を吸収して、紙カードの長さや厚さが増してしまう。逆に湿気の少ない所では、紙カードの湿気が減り紙カードが縮む。どの場合もコンピュータへの入力が困難になるので細心の注意が必要である。
- (2) 紙カードを、カード・キャビネットに保管するとき、止め金できちんと押えておく習慣をつけるように指導する。

2 紙 テ ー プ

2.1 紙テープの作成

目 標

紙テープせん孔機の扱い方を説明し、データを紙テープにせん孔する方法を理解させる。

内 容

次の各項目を説明する。

- (1) 紙テープせん孔機の各部の機能
 - (a) せん孔部
 - (b) フィード・チェック部
 - (c) テープ送り部
 - (d) 紙テープせん孔タイプライタ部
- (2) タイプライタのキーボードの操作の仕方
- (3) 紙テープのセットの仕方
- (4) 用紙のセットの仕方
- (5) 紙テープへのせん孔の仕方
- (6) せん孔中に誤せん孔に気付いたときの処理
- (7) 複写の操作

せん孔された紙テープの内容を新しい紙テープにそのまません孔したり、その一部を変更してせん孔したりする操作の仕方を説明する。

指導上の留意点

- (1) 最初に紙テープせん孔機を見せて、一連の操作を実演してみせる。
- (2) 紙テープのせん孔位置と文字の対照表（コード表）を見せて、この表を参照すれば紙テープが読めるように指導する。表を覚えてしまう必要はない。
- (3) 図や写真を多く活用する。
- (4) キーボードの位置を覚えさせる必要はない。

- (5) 次のようなことはきちんと処理できるように指導する。
- (a) 紙テープの整理整頓、紙テープを捨てる箱などを用意すること。
 - (b) 紙テープせん孔機の周囲の整理整頓
 - (c) 使用前のスイッチの入れ方、使用後のスイッチの切り方
 - (d) タイプライタ用紙の処理
- (6) 紙テープの先端からすぐにデータを入れないこと、せん孔し終った紙テープを切るとき、少し余白部分を残すことに注意する。

2.2 紙テープのチェック

目 標

紙テープ上のデータのチェックの仕方を習得させる。

内 容

コンピュータを使ってチェックする方法は、紙カードのところを参照する。ここでは、その他のチェックの方法を説明する。次のようなことを中心に述べればよい。

- (1) 紙テープの内容を印字してみることによって検査する方法
- (2) 紙テープ検孔機を使って検査する方法

誤せん孔の紙テープを訂正するには、誤せん孔部分を切取って、正しいせん孔テープを継ぎ足さなければならない。次の3つの継ぎ方を説明する。

- (1) 重ね継ぎ
- (2) 継張紙による方法
- (3) 接着テープによる方法

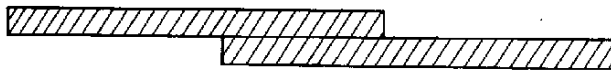
指導上の留意点

紙カードのチェックの項で述べた指導上の留意点(1), (3), (4)はここでも該当する。紙テープのつぎはぎの仕方は、紙テープの種類や機種によって異なるので、各機種の操作説明書に従う必要がある。次の例は、教え方の1例である。

- (1) 継張紙や接着テープを使って継ぐときは、テープ・スプライサを使用する。手でも張れるが注意力を要する。
- (2) 手で紙テープを張合わせるときには、2枚の紙テープの穴の位置がずれないように注意し、

張り合せ部分はサイドのバリを取るため少し切取るようにする。

- (3) 重ね継ぎのとき、テープの進行方向が上になるようにして、継ぎ目部分の長さは2字分以下にする。



←テープ進行方向

- (4) 継張紙で継ぐとき、継張紙が上になるようにし、継ぎ目部分の長さは4～6字分とする。テープ・スプライサを用いるときは、6字分とする。
- (5) 接着テープで張合わせるとき、接着テープが上になるようにする。

2.3 紙テープの取扱い上の注意

目 標

紙テープを取扱うとき注意しなければならないことを理解させる。

内 容

次のことを述べる。

- (1) 紙テープは長いので、使用しないときは、巻いておくこと。
- (2) 折ったりしないこと。
- (3) 何のプログラムか、またはデータであるかがわかるように、紙テープの先端にラベルなどを張りつけて表題をつける。薄い色の紙テープならば、直接紙テープに記入してもさしつかえない。
- (4) 花文字の利用

紙テープの内容を示す文字や模様を紙テープの最初の部分に、眼で読取れる形式にしてせん孔する。

指導上の留意点

1.3 紙カードの取扱い上の注意の項で述べた指導上の留意点はここでも該当する。

2.4 紙テープの保管方法

目標、内容、指導上の留意点とともに1.4 紙カードの保管方法を参照する。

参考文献

1. MODERN DATA PROCESSING

ROBERT R. ARNOLD

HAROLD C. HILL

AYLMER V. NICHOLS

JHON WILEY & SONS, INC.

1969 3,580円

第6章 PUNCHED CARD RECORDING FUNCTIONS

が参考になる。

2. 図説電子計算機 ハードウェアの構成と動作

日本電信電話公社編

電気通信協会

昭45 1,600円

第6章 周辺装置で、紙テープと紙カードについて触れている。

3. 電子計算機とデータ処理

R. N. シュミット

W. E. メイヤーズ 共著

鵜沢昌和 訳

東洋経済新報社

昭44 1,600円

第9章 入力、出力、記憶媒体のところが参考になる。

4. 電子計算機 (1)

金田 弘 著

コロナ社

昭45 1,200円

3. 入出力と補助記憶の入出力装置のところが参考になる。

5. 電子計算機概論——計算機システムと情報処理——

池田敏雄 編

オーム社

昭45 1,000円

第4章 周辺装置の4.3 入出力装置が参考になる。

6. 紙テープ装置

富士通㈱ 実費

紙テープ取扱い上の注意や保存条件に触れている。

7. キーパンチ・オペレータのガイド・ブック

日本アイ・ビー・エム㈱ 実費

けん盤せん孔機の操作の仕方などを詳細に述べている。

第 4 部 コンピュータの運用

ここでは次の用語を教える。

操作（オペレーション）、カード・ジャム、カード・スタッカ、カード・デッキ、カード・ファイル、カード・ホッパ、更新する、再開（再起動）、実行（ラン）、順序検査、ジョグル、ジョブ、チェックポイント、パッチ（切り張り）、保守、補助操作、メッセージ、ライブラリ

教育の目標

コンピュータは、機種によって構成や機能が違うので、操作の仕方も個々のコンピュータによってかなり異っている。しかし、それと同時に、どのコンピュータを使うとしても、共通して考えておかなければならない問題がいくつかある。ここでは、コンピュータの操作について一般的に考えなければならぬ共通の問題を理解させる。プログラミングの実習をするためには、実習に使うコンピュータの固有の操作も知らなければならないが、固有の操作についてはコンピュータ・メーカーが発行している各装置の操作説明書を使って補助説明をすればよい。主として次のような内容について、生徒の理解を深める。

データの取扱い、制御卓の操作、入出力装置の操作、操作の技法、プログラムの管理、プログラムの保守。

内容のあらまし

教える項目と時間数は、次のとおりである。

内 容	時 間 数
コンピュータの構成と環境	4
データの取扱い	8
制御卓の操作	9
入出力装置の操作	12
コンピュータ操作	9
プログラムの保守	8

内 容

1. コンピュータの構成と環境

コンピュータは、中央演算処理装置、入出力装置、補助記憶装置などいくつかの装置から構成されている。それらの装置が、操作する立場からみて調和のとれた構成になっていること、操作（オペレーション）しやすい配置になっていることが大切である。

次のような点について簡単に説明する。

- (1) 操作員が無理なく動けるように各装置の配置を考える。
- (2) システムが拡張できるように配慮しておく。
- (3) 作業の指示 入出力データの整理、デバッグ、紙カードのせん孔など、コンピュータ操作に付随して生ずる作業ができるスペースを確保する。
- (4) 温度調整の仕方。
- (5) 湿度調整の仕方。

温度、湿度が許容量を越えたときの処理についても説明する。

- (6) ごみやちりの処理

2 データの取扱い

入力データの取扱い、出力データの取扱い、データ・ファイルやプログラム・ライブラリなどの扱い方について説明する。

- (1) 入力データの取扱い。

入力データが作られるまでの過程について、第3部データ準備の復習をする。

紙カードや紙テープにせん孔されたデータは、紛失したり、破損されたりすることのないよう十分注意して、コンピュータ室へ運び、処理されなければならない。現実には入力データの紛失や破損がしばしば起っているので、どのような場合にそのようなことが起るか具体的な例をあげて説明するとよい。

事故を防ぐための配慮、入力データの受け渡しの仕方、処理済みの入力データの保管、入力データを訂正するときの注意などについて、考え方を説明する。

- (2) 出力データの取扱い。

印刷用紙に出力される場合と、紙カード、紙テープ、磁気テープなどに出力される場合とに分けて説明する。

印刷する場合、複数個所に配布が必要なときはカーボン付きの複写用紙を使うことができるこ

とを述べる。複写用紙を切離したり、連続用紙を切断したりするための装置について簡単に説明する。

紙カードや紙テープへの出力については、入力データの取扱いに準じて考えてよい。ただし、紙カードに出力する場合は、内容が紙カード上に印字されないで、せん孔された紙カードにマジック・インクなどを使って内容を示す表示を記入する必要がある。紙カード翻訳印字機（インタプリタ）を使えば、紙カードの内容を紙カード上に印字できる。

磁気テープへの出力については、次のデータ・ファイルの取扱いとあわせて説明する。

(3) データ・ファイルの取扱い。

データ・ファイルを記録する媒体としては、紙カード、紙テープ、磁気テープ、磁気ディスクなどがある。それぞれの媒体に記録されているファイルについて、保管の問題を中心に説明する。

次のような点を説明する。

(a) 保管の仕方と保管場所

火災や事故の対策にも触れる。

(b) 保管する期間

(c) 保管の手続き

(d) 使用するときの手続き

(e) 廃棄するときの手続き

磁気テープの保管に関連して三代保管（three generation）の考え方を説明する。

磁気テープ・リールにつける外部表示ラベルについて説明する。

(4) プログラム・ライブラリの取扱い。

プログラム・ライブラリもデータ・ファイルと同じように、紙カード、紙テープ、磁気テープ、磁気ディスクなどに記録されるが、データ・ファイルに準じて扱えばよい。しかし、次のような点は、プログラム・ライブラリ独特の問題なので、特別な配慮が必要である。

(a) しばしば修正されるので元本の保管がむずかしい。

(b) 1枚でも順番がくったり、紛失したりすると使えなくなる。

(c) 文書が完備していないと内容がつかめない。

3 制御卓の操作

制御卓の概要を説明する。細かい点は、コンピュータによって異なるので、それぞれのコンピュータに固有の操作説明書を使って説明するとよい。

例えば次のような点に触れる。

(1) 制御卓のキーやライトの説明

- (a) 電源スイッチ
- (b) 非常用スイッチ
- (c) コンピュータを始動させるためのキー、再始動のためのキー
- (d) コンピュータの実行を停止させるキー (stop key)
- (e) 入出力装置のスタート、ストップ・キー
- (f) プログラム・ロード・キー
- (g) 操作モードを決めるスイッチ

実行モード、表示モード、記憶域のダンプ、アドレス・ストップなどの機能とあわせて説明する。

- (h) アドレス・セットのダイヤルとスイッチ
- (i) エラー・リセット・キー
- (j) レジスタの内容を示すライトとキー
- (k) ロジック・ライト

演算エラー、算術演算オーバーフロー、比較結果などを示すライト、入出力装置への注意を促すライトなどを説明する。

(2) 制御卓の操作の手順

制御卓は、操作員がコンピュータの操作をするときに、一番重要な装置である。操作員がプログラムの操作説明書に従って**実行(ラン)**を行なう手順に添って操作の仕方を説明する。

(a) プログラマから操作員への指示

プログラミングをした人が操作も行なうということは少ない。一般的にはプログラミングとコンピュータの操作とは分業化されていることを述べて、プログラマと操作員の間で作業の仕方について、適切な情報の交換が必要なことを説明する。操作員は、操作に先だって次のようなことをつかんでいなくてはならない。

- ・ 使用するプログラム
- ・ 使用する入出力装置と入出力媒体
- ・ 使用する入力データ
- ・ 使用するデータ・ファイル
- ・ 使用する印刷用紙と印刷の書式

これらの情報はプログラムの操作説明書に記入する。操作説明書の内容と役割を説明する。

(b) 操作員の操作

次のような点を説明する。

プログラムのローディング

使用する入力データやデータ・ファイルのセット

出力媒体のセット

コンピュータの誤動作，入出力エラー，操作員の操作エラー，プログラム・エラーなどを起したときの措置を説明する。エラー処理と関連させて，プログラムの再開（再起動）についても触れる。

(c) タイプライタ型制御卓の操作

タイプライタ型制御卓の役割について次のような点を説明する。

コンピュータと操作員の間の情報交換。

制御卓へのメッセージの印刷，制御卓からの情報の入力。

タイプライタ型制御卓を使って，コンピュータで処理した作業の記録をとることができる。

実行されたジョブの番号，ジョブの名前，作業開始時間，作業終了時間などをタイプライタ型制御卓に打出す仕組みについて簡単に述べる。

4 入出力装置の操作

次のような入出力装置の操作について説明する。

- ・ カード読取装置
- ・ カードせん孔装置
- ・ 紙テープ読取装置
- ・ 紙テープせん孔装置
- ・ 印刷装置
- ・ 磁気テープ記憶装置
- ・ 磁気ディスク記憶装置

(1) カード読取装置の操作

カード・ホッパ，カード・スタッカ，読取機構などカード読取装置の操作に関係の深い部分の説明をする。カード・デック，カード・ファイルという用語がよく使われるので，ここで説明しておく。

カード読取装置の操作に関連して次の説明をする。

- (a) カード・デックのジョゲル
- (b) カード・デックにのせるウェート（おもし）
- (c) カード読取装置の始動の仕方
- (d) 紙カードのフィード・エラー（ニック）
- (e) カード・ジャム
- (f) ジャム・カードの取出し方
- (g) 最終カードの読込ませ方

紙カードは、紙テープや磁気テープなどと違って、紙カード1枚ごとにデータが分離しているので、順序がくるったり、紛失したりすることが起りやすい。そのような誤りを発見する手段として一般的に行なわれている順序検査（シーケンス・チェック）とトータル・チェックについて説明する。

(2) カードせん孔装置の操作

カードせん孔装置は、カード読取装置に準じて考えればよい。

カード・ホッパには、ブランク・カードを置く。せん孔された紙カードの最後の1枚の取出し方、せん孔エラーが起ったときの処理などは、カード読取装置と異なるので説明しておく。

(3) 紙テープ読取装置

紙テープ読取装置には、高速読取装置と、卓上型の低速読取装置とがある。個々の操作は、装置によって異なるので、必要に応じて、各装置の操作手引書を参照する。

一般的な説明としては、次のような程度の説明をすればよい。

(a) 紙テープの読取り方式

ストリップ・フィード

センタ・ロール・フィード

リール・フィード

(b) 読取りエラーを起したときの処理

(c) 処理後の紙テープの後始末

(4) 紙テープせん孔装置

必要に応じて、第3部データ準備の紙テープの項と上記の紙テープ読取装置の場合を参照する。

(5) 印刷装置

印刷装置には、ライン・プリンタ（行印字機）と、タイプライタ型の印刷装置とがある。

ライン・プリンタについては、次のような点を説明する。

(a) キーとライト

スタート・キー
ストップ・キー
キャリッジ・ストップ・キー
エラー・リセット・キー
スキップ・キー
スペース・キー
シングル・サイクル・キー
プリント・レディ・ライト
プリント・チェック・ライト
エンド・オブ・フォーム・ライト

(b) 用紙のセットの仕方

(c) 印刷位置の調整の仕方

(d) 印刷の濃さの調整の仕方

(e) キャリッジ・テープの作り方と、キャリッジ・テープのセットの仕方

テープ・パンチの使い方も説明する。

ライン・プリンタについては、チェーン・プリンタが広く使われているので、チェーンと文字セットについても簡単に触れるとよい。

タイプライタ型印刷装置については、タイプライタ型制御卓を参照し、次の説明をする。

(a) 用紙のセットの仕方

(b) キャリッジ・リターン

(c) ライン・フィード

(d) バック・スペース

(6) 磁気テープ装置

最初に磁気テープの機構を簡単に説明する。キーとスイッチのセットについて、次のような点を説明する。

(a) リセット・キー

(b) ロード・キー

(c) リワインド・キー

- (d) スタート・キー
- (e) ストップ・キー
- (f) アンロード・キー
- (g) 装置アドレスのセット

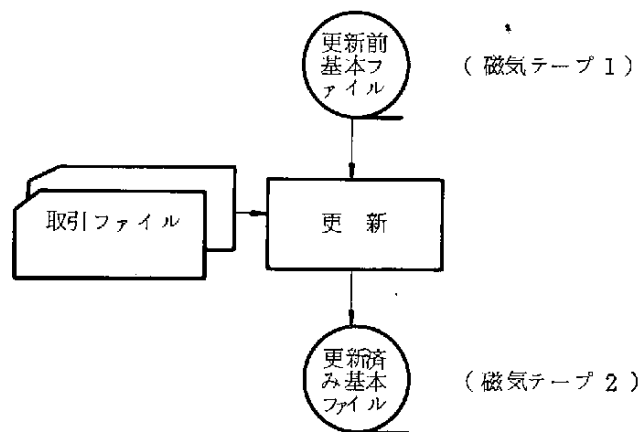
磁気テープ・リールの取扱い方について次の点を説明する。

- (a) テープ・リールの持運び方
- (b) ファイル保護リングの扱い方
- (c) 磁気テープの外部表示ラベルの付け方
- (d) 磁気テープが破損したときの処理
- (e) ロード・ポイント・マーカの付け方
- (f) 磁気テープ・リールの保管の仕方

磁気テープ記憶装置の操作について、次の点を説明する。

- (a) 磁気テープ・リールの取付け方
- (b) 磁気テープ記憶装置とレディの状態にするための操作
- (c) 読み書きヘッドの掃除
- (d) 読み書きエラーがひんばんに起ったときの処理

磁気テープ記憶装置では、ファイルを更新するときの操作が重要である。次のような図を示してファイルを更新するときの操作の仕方を説明することが望ましい。



磁気テープ 1 については、次のような処理が必要である。

- (a) ファイル保護リングが取りはずされていることを確認する。

(b) 外部表示ラベルを見て内容を確認する。

(c) 磁気テープ記憶装置にセットする。

磁気テープ2については、次のような処理が必要である。

(a) 外部表示ラベルを見て、磁気テープの内容が消されてもよいものかどうかを確認する。

(b) ファイル保護リングを取付ける。

(c) 磁気テープ記憶装置にセットする。

(d) 処理のあと、ファイル保護リングを取りはずす。

(e) 外部表示ラベルを付けて、作成日、作成者、ファイル名、内容などについての表示を記入する。

(7) 磁気ディスク記憶装置

磁気ディスク記憶装置は、簡単に触れる程度でよい。次の2点について説明する。

(a) 磁気ディスク・パックの取扱い

磁気ディスク・パックの持運び方、保管の仕方、外部表示ラベルの付け方などを説明する。

(b) 磁気ディスク記憶装置の操作

磁気ディスク・パックのセットの仕方、磁気ディスク記憶装置のレディの仕方、読み書きエラーがひんばんに起きたときの処理などについて述べる。

磁気ディスク・パックは、一般に破損したとき内容を復元することが困難なので、内容を複写して保管しておくことが必要である。磁気テープに内容をダンプして保存する方法や、ダンプのサイクルなどについて説明するとよい。

(8) 補助操作

コンピュータで処理を行なうとき、コンピュータには直接接続していない補助装置を使うことがある。けん盤せん孔機や検孔機も補助装置の一種である。

その他によく使われる補助装置として、カード分類機やカード翻訳印字機などがある。これらの装置の機能や操作について簡単に触れる。

5 コンピュータ処理

コンピュータを使って処理を行なうときに持っていなければならない基礎的な知識について教える。次のような点を説明する。

コンピュータの使用計画(スケジューリング)

作業記録と故障時の処理

操作の技法

(1) コンピュータの使用計画

コンピュータで作業を行なうときは、コンピュータの使用計画をたてることが非常に重要である。使用計画をたてるときに、考慮しなければならない点について説明する。次のような点を述べる。

(a) 実行時間の推定

(b) 人手による処理の時間

紙カード、紙テープ、印刷用紙、磁気テープ・リールのセットなど、操作員が操作に要する時間

(c) リラン(再操作)についての考慮

入力データの誤りや操作の誤りなどで、リランが必要になることが多い。

(d) 操作開始可能時間の推定

データの到着時間やデータのエラー・チェックに要する時間などを考慮する。

(e) 要求されているターンアラウンド時間

作業の優先度や緊急度を考える。

(f) プログラム・テスト時間の推定

プログラム・テストをするのに必要なコンピュータの時間を推定する。推定の仕方にはいろいろな方法があるので、おおまかに見当を付ける程度でよい。

(2) 作業記録と故障時の処理

作業記録簿の実例を紹介し、作業記録簿の付け方を説明する。コンピュータの故障はいろいろな場合が考えられるので、故障したときの処理については、実務を通して経験的に学ぶより仕方のないことが多いが、ここでは、一般的にいえることがらとして次のような点を述べる。

(a) 故障の確認

操作の誤りと故障を見違えることが多い。

(b) 故障の状態を正確に記録する。

故障に関連した資料を保存する。

(c) 故障したときの連絡の仕方

(d) 故障時の代替処理

緊急な作業については、別な処理の仕方を考えておかななくてはならない。

(3) 操作の技法

次のような点について説明する。

プログラムのパッチ（切り張り）

記憶ダンプ

チェックポイントと再開

ユーティリティ・プログラム

オペレーティング・システム

(a) プログラムのパッチ（切り張り）

紙カードにせん孔されている実行プログラムのパッチの仕方、主記憶にロードされているプログラムのパッチの仕方などを簡単に説明する。

(b) 記憶ダンプ

記憶ダンプをとるときに操作の仕方を説明する。制御卓のキーの操作で行なう方法と、ユーティリティ・プログラムによる方法とがある。

(c) チェックポイントと再開

チェックポイントの概念を説明する。

チェックポイントをとることが、なぜ重要かを説明する。磁気テープ・ファイルを入力して集計しながら印刷する処理を長時間続ける場合などを例にあげて説明するとよい。

チェックポイント・レコードを使って、処理の再開をはかるときの方法を簡単に説明する。チェックポイントを考えたプログラムの例をあげにくければ、分類併合プログラムを取上げて説明するとよい。

(d) ユーティリティ・プログラム

紙カードから磁気テープへの変換、磁気テープの印刷、磁気テープの複写、磁気ディスクから磁気テープへのダンプなど、よく使われるものの中から1、2取上げて操作を説明する。

(e) オペレーティング・システム

オペレーティング・システムの制御のもとで処理を行なうときの特徴を簡単に説明する。

連続処理、並行処理、オンライン処理などの操作について、特徴的なことがらを取上げて説明すればよい。

6 プログラムの保守

プログラムは絶えず性能の向上や機能の拡張が求められているので、しばしば変更されること

になる。ここでは、プログラムを変更するときに考慮しなければならない点や、保守の技法などについて説明する。

次のような点を説明する。

(a) プログラムの保守をするときの手続き

プログラムを安易に修正すると混乱を起すものになる。特に、プログラム作成や保守の担当者と操作員が異なる場合には、プログラムを修正したときの連絡の不徹底が、重大な結果を招くことがよくある。プログラムの保守の手続きをどうしたらよいか例をあげて説明する。

(b) 文書化

プログラムを修正すると、それにもなってプログラムの仕様書や説明書などを修正しなくてはならない。プログラムについての文書と、プログラムそのものがずれているときに生ずるむだや混乱について十分理解させておく必要がある。

(c) プログラムの管理

プログラム改訂のレベルがすぐわかるようにしておくこと、事故に備えて複写をとっておくことなどが重要である。プログラムの改版番号(version No)や改訂番号(modification No)について説明する。

(d) モジュール構造

プログラムの改訂を容易にする技法として、プログラムのモジュール構造について説明する。モジュール構造の技法を詳しく説明する必要はない。プログラムの設計をするときに、保守を容易にするという考慮が必要であることを理解させることに重点をおいて説明するとよい。給与計算のプログラムなどを例にあげて、保守の負担を少なくするためにどうしたらよいかを生徒に考えさせてみるのもよい。

(e) プログラムの互換性

プログラムの修正の仕方によっては、改訂する前に使っていたデータや他のプログラムが、新しいプログラムでは利用できないことがある。そうした例をあげて、互換性の問題を考えさせる。

プログラミング・システムが改訂されることによって、プログラムを書直さなければならない例や、データの形式を変えることによって生ずる混乱などを説明する。間接的に互換性を持たせるためのデータ変換プログラムについて触れる。

指導上の留意点

コンピュータの操作には、特定の機種に限定されない一般的なことがらと、個々の機種に固有

な操作の問題とがある。機種に固有のことからは、それぞれの装置の解説書を使って必要に応じて説明したり、実習を行なう。一般的なことがらとして、特に重要なのは操作の技術よりも、操作の心構えである。

操作をするときに重要なことは、いかにして効率的に、正確に操作を行なうかということである。したがって、説明が技術的な問題に片寄らないようにすることが望ましい。

コンピュータ操作の実習は、プログラミングの実習とあわせて行なってもよいし、独自に行なってもよい。

実習としては、紙カード、紙テープ、印刷用紙、磁気テープなどをセットするという簡単なもののほか、次のようなものが考えられる。

(1) 制御卓の操作

制御卓から、データを主記憶にそう入する。

主記憶の内容をダンプする。

制御卓の操作で命令を1ステップずつ実行させる。

(2) ユーティリティ・プログラムの使用

制御カードを作らせ、制御カードを使ってユーティリティ・プログラムで処理を行なう。

(3) 操作説明書の作成

プログラミングの実習で作成したプログラムの操作説明書を作成させる。

コンピュータ室は、温度、湿度が一定に保たれていることや、清潔であることが重要である。生徒には実習の前後の整理・整とんに十分気をつけさせる。

コンピュータの環境について、絶えず気を配るように指導する。

できるだけ実物を見せて説明する。

スライドや写真もできるだけ多く使った方がよい。

参考文献

1. MODERN DATA PROCESSING

ROBERT R. ARNOLD

HAROLD C. HILL

AYLMER V. NICHOLS

JOHN WILEY & SONS, INC.

1969 3,580円

第17章 ELECTRONIC DATA PROCESSING OPERATIONS が参考になる。

2 電子計算機の原理と構造

電気学会通信教育会 著
社団法人 電気学会 発行
取次販売所 桐学献社
昭45.9 1,000円

第2章 小型電子計算機、第3章 高性能の電子計算機が参考になる。

3 だれにも使えるコンピュータ

南条 優 著
日本生産性本部
昭44.6 480円

I コンピューターの基礎知識の「コンピューターのしくみ」が参考になる。

4 初等電子計算機読本

石井康雄 著
オーム社
昭41.7 550円

第2章 入出力装置が参考になる。

5 自己学習 コンピュータの基礎

M. V. ファリナ 著
藤沢 忠 訳
ダイヤモンド社
昭45.4 850円

第14章 磁気テープ・ファイルや、第15章 ファイル維持、第16章 コンピュータ・センターが参考になる。

6 コンピュータ科学入門

涌田宏昭 編
白桃書房
昭44.1 800円

第4章 コンピュータ・システムの構成、第5章 プログラミングの話(その1),

第5章 プログラミングの話(その2)が参考になる。

7 磁気テープ読本

木沢 誠 編

データ・ネット社

昭44.12 800円

磁気テープ全般について詳しく述べている。

8 電子計算機とデータ処理

R. N. シュミット 著

W. E. メイヤーズ

鶴沢昌和 訳

東洋経済新報社

昭44.9 1,600円

第17章 システムと手続きが参考になる。

9 電子計算機(1)

金田 弘 著

コロナ社

昭45.9 1,200円

4. オペレーティング・システム, 6. データ処理の計画と設計が参考になる。

10 各コンピュータ・メーカーの機械操作手引書

実費

第 3 編 付 録

本編は、「初級情報処理技術者育成指針」を活用するにあたり、参考となる事項をとりまとめたものでありますが、一部について外部資料を引用させていただきました。ここに、内容の転載について快くご許可いただいた下記の諸機関・団体に対し感謝の意を表します。

(転載順)

通商産業省重工業局情報処理振興課

社団法人日本電子工業振興協会

財団法人日本規格協会

通商産業省工業技術院標準部

文部省初等中等教育局職業教育課

財団法人日本経営情報開発協会

付録 1 情報処理技術者試験の概要

目 次

1. 昭和45年度情報処理技術者試験案内書	232
2. 情報処理振興事業協会等に関する法律	238
3. 情報処理技術者試験規則	241

昭和44年度から始った情報処理技術者試験について、通商産業省重工業局情報処理振興課から発行された昭和45年度情報処理技術者試験案内書の抜粋と、関係法規を示す。

1. 昭和45年度情報処理技術者試験案内書 (抜粋)

I 情報処理技術者試験について

昨年度発足した情報処理技術者認定試験は、今年度から情報処理振興事業協会等に関する法律第6条(別項参照)に基づく情報処理技術者試験として行なわれることになりました。

情報処理技術者試験の目的は、

- (1) 企業、団体、官公庁等に勤務する情報処理技術者等に目標を示し、刺激を与えることによってその技術の向上を図ること。
- (2) 情報処理技術者として備えるべき能力についての水準を示すことにより、教育水準の確保に資すること。
- (3) 電子計算機を利用する企業、団体、官公庁等で情報処理技術者の採用、配置、昇任等を行なう際に役立つよう客観的な評価の尺度を提供し、これを通じて情報処理技術者の社会的地位の確立をはかること。

などにあります。したがってこの試験は、資格試験ではなく、受験資格にも制限はありません。

今年度から行なわれる情報処理技術者試験の区分、試験科目等その大要は、昨年度行なわれた情報処理技術者認定試験とは同じです。

なお、参考のため昨年度行なわれた試験の結果を示すと次のとおりです。

昭和44年度応募者数、受験者数および合格者数一覧

区 分	応 募 者	受 験 者	受 験 率	合 格 者	合 格 率	合格者の平均年齢
第1種	12,924人	10,527人	81.5%	811(22)人	7.7%	28.1歳
第2種	29,098	22,057	75.8	1,832(177)	8.3	24.6
合 計	42,022	32,584	77.4	2,643(199)	8.1	25.7

()内は女性で内数

昭和44年度合格者の学歴別、職業別内訳

区 分		第 1 種	第 2 種	合 計
学 歴 別	大 学	人 682 (84.2) %	人 967 (52.8) %	人 1,649 (62.3) %
	短 大	12 (1.5)	66 (3.6)	78 (3.0)
	高 校	99 (12.1)	773 (42.1)	872 (33.0)
	中 学	3 (0.4)	10 (0.5)	13 (0.5)
	不 明 ・ そ の 他	15 (1.8)	16 (0.9)	31 (1.2)
	合 計	811 (100.0)	1,832 (100.0)	2,643 (100.0)
職 業 別	会 社 員	702 (86.6)	1,385 (75.6)	2,087 (79.0)
	公 務 員	47 (5.8)	126 (6.9)	173 (6.5)
	学校・研究機関勤務者	9 (1.1)	31 (1.7)	40 (1.5)
	学 生 ・ そ の 他	18 (2.2)	172 (9.4)	190 (7.2)
	不 明	35 (4.3)	118 (6.4)	153 (5.8)
合 計		811 (100.0)	1,832 (100.0)	2,643 (100.0)

(注) 学歴別区分の大学には大学院、および旧制高専を含む。

Ⅱ 昭和45年度情報処理技術者試験および受験手続きについて

1. 試験の区分とその対象

情報処理技術者試験は、下に示すように、第1種情報処理技術者試験と第2種情報処理技術者試験の2つに区分して行なわれ、試験を受けようとする者の年齢・性別・職歴は問いません。

(1) 第1種情報処理技術者試験

情報処理技術者のうち、プログラムの設計、高度のプログラムの作成および第2種情報処理技術者等の指導に主として従事する者、すなわち大学卒業程度の一般常識を有し、3年程度以上のプログラミング経験を有するシニアプログラマを想定して試験が行なわれます。

(2) 第2種情報処理技術者試験

情報処理技術者のうち、プログラム設計書に基づくプログラムの作成に主として従事するもの、すなわち高等学校卒業程度の一般常識を有し、1年程度以上のプログラミング経験を有する一般プログラマを想定して試験が行なわれます。

2. 試験日および試験地

試験は、第1種情報処理技術者試験、第2種情報処理技術者試験とも昭和45年1月8日(日)

曜)に行なわれます。試験の時間割および試験地は以下のとおりです。

(1) 試験の時間割

試験の区分	午前(10時～12時)	午後(1時30分～4時)
第1種情報処理技術者試験	多枝選択式	記述式
第2種情報処理技術者試験	多枝選択式	記述式

(2) 試験地

札幌市、仙台市、東京都、名古屋市、大阪市、広島市、高松市および福岡市の全国8都市

3. 試験の方法

試験は、多枝選択式および記述式による筆記試験により行なわれます。なお、試験会場に持込み可能なものは以下のものに限られ、参考書、マニュアル等は使用できません。

(1) BまたはHBの鉛筆および鉛筆削り

(2) 消しゴム

(3) フローチャート用テンプレート(JIS規格のものが望ましい)または直定規

(4) 時計

4. 受験申込みに必要な書類

(略)

5. 受験案内書および受験願書用紙の請求

(略)

6. 受験願書の提出先

(略)

7. 受験願書の受付期間

(略)

8. 受験票および受験案内

(略)

9. 合格者の発表

合格者の受験番号および氏名は、昭和46年2月中(受験者が多数にのぼる場合は3月になることもあります。)に官報に公示します。

また合格者に対しては、情報処理技術者試験合格証書を交付します。

10. その他

(略)

Ⅲ 試験科目およびその範囲

1. 第1種情報処理技術者試験

(1) ソフトウェアの知識(ウェイト25%)

- ① 電子計算機による問題処理手順、フローチャート(流れ図)、プログラム記憶方式等に関すること。
- ② ファイルに関すること。
- ③ プログラム言語(特殊問題向きプログラム用言語を含む。)および汎用プログラムパッケージに関すること。
- ④ オペレーティング・システムの基本概念に関すること。
- ⑤ オンライン方式、タイムシェアリング方式に関すること。
- ⑥ ドキュメンテーション(文書化)に関すること。

(2) プログラムの設計能力(ウェイト30%)

システム設計書に基づくプログラムに関するものであって、次に掲げるもの。

- ① 入出力データのフォーマットの設計に関すること。
- ② ファイルの設計に関すること。
- ③ コードの設計に関すること。
- ④ プログラム構造の設計(セグメンテーション、モジュール化等を含む。)に関すること。
- ⑤ デバグ(手直し)方式の設計に関すること。

(3) プログラムの作成能力(ウェイト15%)

プログラム設計書に基づくプログラムの作成に関すること。ただし、プログラム用言語については、FORTRAN, ALGOL, COBOL PL/Iおよびアセンブリー言語のうちから受験者が選択するものとする。(注参照)

(4) ハードウェアの知識(ウェイト15%)

- ① 電子計算機を構成する諸装置(入出力装置、制御装置、演算装置、記憶装置等)の種類、機能および特徴に関すること。
- ② データ伝送ならびに端末機器の種類、機能および特徴に関すること。
- ③ システムの構成とその効率に関すること。

(5) 関連知識(ウェイト15%)

- ① 情報処理に関する一般的知識(情報処理用語, 数学, 統計学等)
- ② 事務管理, 経営管理手法ならびに技術計算および数値解析に関することのうち, 受験者が選択する事項。

2. 第2種情報処理技術者試験

(1) ソフトウェアの基礎知識(ウェイト25%)

- ① 電子計算機による問題処理手順, フローチャート(流れ図), プログラム記憶方式に関すること。
- ② ファイルの概念に関すること。
- ③ プログラム用言語(特殊問題向きプログラム用言語を除く。)に関すること。
- ④ 汎用サービスプログラム(入出力変換プログラム, 分類・併合プログラム, 技術計算用ライブラリ等。)に関すること。

(2) プログラムの作成能力(ウェイト45%)

プログラム設計書に基づくプログラムの作成に関すること。ただし, プログラム用言語に関しては, FORTRAN, ALGOL, COBOL, PL/Iおよびアセンブリ言語のうちから受験者が選択するものとする。(注参照)

(3) ハードウェアの基礎知識(ウェイト15%)

電子計算機を構成する諸装置(入出力装置, 制御装置, 演算装置, 記憶装置等)の種類, 機能および特徴に関すること。

(4) 関連知識(ウェイト15%)

- ① 情報処理に関連する一般的知識(情報処理用語, 数学, 統計学等)
- ② データ処理の基礎概念(コード化, データチェック方式, 計算誤差等)に関すること。

注) プログラムの作成能力の科目において出題されるプログラム用言語の水準は次の範囲です。

プログラム用言語名	第 1 種	第 2 種
FORTRAN	JIS水準7000	JIS水準3000
ALGOL	JIS水準5050	JIS水準3030
COBOL	COBOL 65	COBOL 65 ただし, 分類, 報告 書機能, 大記憶等を

		除く。
PL／I	水準は特に規定しない。（第1種，第2種の 差は，問題の難易による。）	
アセンブリー言語	仕様は問題の中で規定する。	

IV 受験願書の作成要領

（略）

2. 情報処理振興事業協会等に関する法律

目 次

第1章	総則(第1条・第2条)
第2章	電子計算機利用高度化計画等(第3条—第6条)
第3章	情報処理振興事業協会
第1節	総則(第7条—第15条)
第2節	設立(第16条—第20条)
第3節	管理(第21条—第27条)
第4節	業務(第28条—第30条)
第5節	財務及び会計(第31条—第36条)
第6節	監督(第37条・第38条)
第7節	補則(第39条—第41条)
第4章	罰則(第42条—第44条)
	付則

第1章 総 則

(目的)

第1条 この法律は、電子計算機の利用及びプログラムの開発を促進し、プログラムの流通を円滑にし、並びに情報処理サービス業等の育成のための措置を講ずること等によって、情報化社会の要請にこたえ、もって国民生活の向上及び国民経済の健全な発展に寄与することを目的とする。

第2条 この法律において「情報処理」とは、電子計算機(計数型のものに限る。以下同じ。)を使用して、情報につき計算、検索その他これらに類する処理を行なうことをいう。

2. この法律において「プログラム」とは、電子計算機に対する指令であって、1の結果を得ることができるように組み合わされたものをいう。

3. この法律において「情報処理サービス業」とは、他人の需要に応じてする情報処理の事業をいい、「ソフトウェア業」とは、他人の需要に応じてするプログラムの作成の事業をいう。

第2章 電子計算機利用高度化計画等

(電子計算機利用高度化計画)

第3条 次に掲げる電子計算機及びプログラムについて、電子計算機利用高度化計画(以下「計画」という。)を通商産業大臣(電子計算機に電気通信回線を接続してする情報処理のために開発するプログラムに係る部分については、通商産業大臣及び郵政大臣、以下この条において同じ。)が定めるものとする。

i 情報処理の振興を図るため利用を特に促進する必要がある電子計算機

ii 情報処理の振興を図るため開発を特に促進する必要がある、かつ、広く利用される種類のプログラム(主としてiの事業の分野における情報処理を目的とするものを除く。)

2. 計画には、電子計算機の設置及びプログラムの開発の目標となるべき事項について定めるものとする。

3. 計画を定めるにあたっては、あらかじめ、関係行政機関の長に協議するとともに、政令で定めるところにより、電子情報処理振興審議会及び郵政審議会の意見をきくものとする。

4. 関係行政機関の長は、前項の協議を受けたときは、関係審議会等の意見をきくものとする。

5. 第1項の規定により計画を定めたときは、通商産業大臣は、その要旨を公表しなければならない。

6. 前3項の規定は、計画の変更について準備する。

(資金の確保)

第4条 政府は、前条第1項第1号に掲げる電子計算機の設置及び同項第2号に掲げるプログラムの開発の促進に必要な資金の確保又はその融通のあつせんに努めるものとする。

2. 前項の措置を講ずるにあたっては、中小企業者に対する特別の配慮がなされなければならない。

(プログラム調査簿)

第5条 通商産業大臣は、円滑な流通を図る必要があると認められるプログラム(主としてiの事業の分野における情報処理に用いられるものを除く。)について、その概要を記載したプログラム調査簿を作成し、これを利用しようとする者の閲覧に供しなければならない。

(情報処理技術者試験)

第6条 通商産業大臣は、情報処理に関する業務を行なう者の技術の向上に資するため、情報処理に関して必要な知識及び技能について情報処理技術者試験を行なう。

2. 情報処理技術者試験を受けようとする者は、政令で定めるところにより、受験手数料を納付

しなければならない。

第3章 情報処理振興事業協会

(以下略)

3. 情報処理技術者試験規則

通商産業省令第59号

情報処理振興事業協会等に関する法律施行令(昭和45年政令第207号)第5条の規定に基づき、情報処理技術者試験規則を次のように制定する。

昭和45年7月13日

通商産業大臣 宮 沢 喜 一

情報処理技術者試験規則

(試験の区分)

第1条 情報処理振興事業協会等に関する法律(昭和45年法律第90号、以下「法」という。)

第6条第1項の規定による情報処理技術者試験(以下「試験」という。)は、次の表の左欄に掲げる試験に区分し、それぞれ同表の右欄に掲げる知識および技能を対象として行なうものとする。

試験の区分	対象となる知識および技能
第1種情報処理技術者試験	プログラムの設計およびプログラムの作成に必要な知識および技能
第2種情報処理技術者試験	プログラムの作成に必要な知識および技能

(試験の科目等)

第2条 試験の科目およびその範囲は、別表に掲げるとおりとする。

2. 試験は、筆記試験により行なうものとする。

(試験委員)

第3条 試験の事務は、通商産業大臣が任命する試験委員が行なうものとする。

(試験手続)

第4条 試験を受けようとする者は、様式第1の情報処理技術者試験願書を、あらかじめ通商産

業大臣が定める試験地を管轄する通商産業局長を経由して、通商産業大臣に提出しなければならない。

（合格証書の交付）

第5条 通商産業大臣は、試験に合格した者に対し、様式第2の情報処理技術者試験合格証書（以下「合格証書」という。）を交付する。

（合格証書の再交付）

第6条 合格証書をよごし、損じ、または失ってその再交付を受けようとする者は、様式第3の情報処理技術者試験合格証書再交付申請書を通商産業大臣に提出しなければならない。

2. 合格証書をよごし、または損じてその再交付の申請をする場合は、前項の情報処理技術者試験合格証書再交付申請書に、当該合格証書を添付しなければならない。

3. 第1項の規定により合格証書の再交付を受けようとする者は、再交付手数料として200円を納付しなければならない。

（手数料納付の方法）

第7条 法第6条第2項に規定する受験手数料および前条第3項に規定する再交付手数料は、収入印紙をもって納付しなければならない。

附 則

この省令は、公布の日から施行する。

別表

試験の区分	試験科目	試験科目の範囲
第1種情報処理 技術者試験	1.ソフトウェア の知識	1.電子計算機による問題処理手順、フローチャート（流れ図）、プログラム記憶方式等に関すること。 2.ファイルに関すること。 3.プログラム用言語（特殊問題向けプログラム用言語を含む。）および汎用プログラムパッケージに関すること。 4.オペレーティング・システムの基本概念に関すること。 5.オンライン方式・タイムシェアリング方式等に関すること。 6.ドキュメンテーション（文書化）に関すること。
	2.プログラムの 設計能力	システム設計書に基づくプログラムの設計に関するものであって、次に掲げるもの 1.入出力データのフォーマットの設計に関すること。 2.ファイルの設計に関すること。 3.コードの設計に関すること。 4.プログラム構造の設計（セグメンテーション、モジュール化等を含む。）に関すること。 5.デバッグ（手直し）方式の設計に関すること。 6.チェック方式の設計に関すること。
	3.プログラムの 作成能力	プログラム設計書に基づくプログラムの作成に関すること。ただし、プログラム用言語については、FORTRAN, ALGOL, COBOL, PL/Iおよびアセンブリ言語のうちから受験者が選択するものとする。
	4.ハードウェア の知識	1.電子計算機を構成する諸装置（入出力装置、制御装置、演算装置、記憶装置等）の種類、機能および特徴に関すること。 2.データ伝送ならびに端末機器の種類機能および特徴に

		関すること。
	5.関連知識	3.システムの構成とその効率に関すること。 1.情報処理に関連する一般的知識。 2.事務管理，経営管理手法ならびに技術計算および数値解析に関することのうち，受験者が選択する事項。
第2種情報処理 技術者試験	1.ソフトウェア の基礎知識	1.電子計算機による問題処理手順，フローチャート（流れ図），プログラム記憶方式等に関すること。 2.ファイルの概念に関すること。 3.プログラム用言語（特殊問題向けプログラム用言語を除く。）に関すること。 4.汎用サービスプログラム（入出力変換プログラム，分類，併合プログラム，技術計算用ライブラリー等）に関すること。
	2.プログラムの 作成能力	プログラム設計書に基づくプログラムの作成に関すること。ただし，プログラム用言語については，FORTRAN, ALGOL, COBOL, PL/Iおよびアセンブリ言語のうちから受験者が選択するものとする。
	3.ハードウェア の基礎知識	電子計算機を構成する諸装置（入出力装置，制御装置，演算装置，記憶装置等）の種類，機能および特徴に関すること。
	4.関連知識	1.情報処理に関連する一般的知識。 2.データ処理の基本概念（コード化，データチェック方式，計算誤差等）に関すること。

受験者台帳		昭和 年 月 日提出	
提出日前6月以内に撮影して撮影した写真 （無地背景の正面上半身像で裏面に氏名・生年月日、試験の区分を記載したもの）をのり付けすること。		総合検査番号 受験番号	
よりかな 氏 名		性 別 男 女 1 2 印	
本都道府県名・コード（裏面参照） 生年 月 日		明治 M 大正 T 昭和 S 年 月 日	
現住所		郵便番号 電話番号	
※住所変更記入欄			
緊急連絡先（緊急の場合に日中でも連絡できるところ（勤務先等）を記入して下さい。）			
（所在地） （名称）		郵便番号 電話番号	
最終学歴		卒業	
1 大 学 院		卒 業	
2 大 学		中 退	
3 計 算 機 専 門 大 学		在 学 中	
4 短 大		他	
5 高 専		高 専	
6 高 専		高 専	
7 中 学		中 学	
8 そ の 他		そ の 他	
勤務先または職業区分			
プログラミングを研修したところ （主なもの1つだけ）			
1 電算機製造または販売会社		1 企業内教育	
2 ソフトウェア会社		2 学校（3を除く。）	
3 計算センター等		3 電算機学校	
4 上記以外の一般企業・団体		4 通信教育	
5 官公庁		5 電算機製造会社等の教育	
6 学校・研究機関		6 テレビ放送講座	
7 自 営		7 各種セミナー	
8 学 生		8 独学	
9 その他		9 その他	
第1種情報処理技術者試験 受験希望者は、下のわく内 を記入しなすこと。		第2種情報処理技術者試験 受験希望者は、下のわく内 を記入しなすこと。	

合格証書送付先
□□□-□□

（現住所）

情報処理技術者試験
受 験 要 領

昭和 年度	
試験の区分	部 類
試験地	
ふりがな氏名	
受験番号	

五
万

(切り離さないこと)

電話	()
※ 合格証書番号	
※ 受験番号	

※印のわく内は記入しないこと

試験当日必ず持参すること。

裏面に受験票、受験案内を受け取る場合の住所、氏名を記載すること。

同封の受験案内をよく読んでおくこと。

封印のわく内は記入しないこと。

提出書類のチェックリスト（すべての記入が終わったときは、下巻の白丸を黒く塗りつぶしながら再度チェックして下さい。）

記入項目	受験者台帳	合格者台帳 見付表	受験票	受験番号	記入場所
試験の区分(第1種または第2種)			○ 表裏	○ 表裏	2
受験地(全図8か所の内いずれか1か所)			○	○	2
本番(都道府県名のみ、受験者台帳にはコードも記入)	○				2
現在所・郵便番号(土地、アパート、○○様方を明確に)	○	○	○	○	4
氏名・ふりがながなく(横書で正確に記入)	○	○	○	○	6
印	○			○	2
生年月日	○			○	2
写真(のり付けすること)	○		○		
1,000円の入金印紙(のり付けすること)				○	1

情報処理技術者試験
受 験 願 書

受 付 年 月 日	
--------------	--

通商産業大臣

12

昭和 年 月 日提出

ふりがな
氏 名 印

希望する 試験の区分	第 種
希望する 受 験 地	
木 籍 (都道府県名)	
生 年 月 日	
※ 整理番号	

紙 印 入 収
1,000円

(消印しないでの
り付けすること)

裏面に住所・氏名を記載すること。
※印のわく内は記入しないこと。

記載注意

- 1.該当事項を記入または該当する記号・番号を1つだけ○で囲むこと。
- 2.記載事項は楷書で正確に記入すること。
- 3.※印のわく内には記入しないこと。
- 4.勤務先または職業区分欄のうち
 - ③計算センター等は、データバンク等の情報処理サービス業者を含む。
 - ⑧学生は、職業を有しない者のみ
 - ⑨その他は、無職および分類不能の勤務先・職業として記入すること。
- 5.都道府県コードは2けたのアラビア数字を用いて次表により記入すること。

01 北海道	11 埼玉県	21 岐阜県	31 鳥取県	41 佐賀県
02 青森県	12 千葉県	22 静岡県	32 鳥根県	42 長崎県
03 岩手県	13 東京都	23 愛知県	33 岡山県	43 熊本県
04 宮城県	14 神奈川県	24 三重県	34 広島県	44 大分県
05 秋田県	15 新潟県	25 滋賀県	35 山口県	45 宮崎県
06 山形県	16 富山県	26 京都府	36 徳島県	46 鹿児島県
07 福島県	17 石川県	27 大阪府	37 香川県	47 沖縄県
08 茨城県	18 福井県	28 兵庫県	38 愛媛県	48 外国
09 栃木県	19 山梨県	29 奈良県	39 高知県	
10 群馬県	20 長野県	30 和歌山県	40 福岡県	

受験票送付先

□□□-□□

(現住所)

様方

殿

電話番号()

試験終了後も合格者の発表まで大切に保存して下さい。

様方

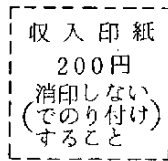
() 電話

氏名

(現住所)

郵便番号

様式第三



※受理年月日	
※再交付年月日	

情報処理技術者試験合格証書再交付申請書

通商産業大臣

殿

昭和 年 月 日提出

氏名 印

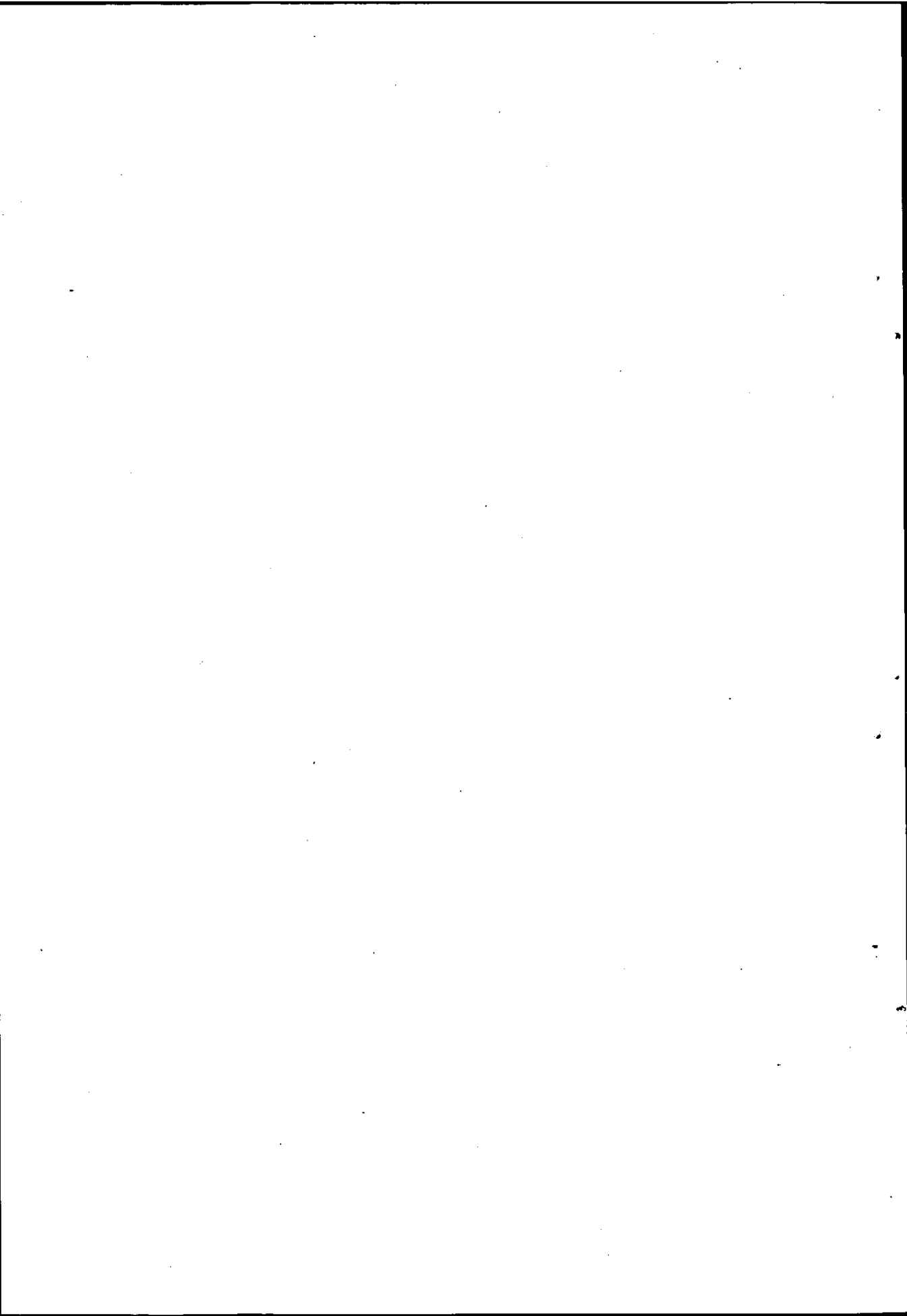
本籍(都道府県名)	
現住所(郵便番号)	
生 年 月 日	年 月 日
受験した年	昭和 年
受験した試験の区分	第 種情報処理技術者試験
合格証書の番号	第 号
再交付を受ける理由	

- 備考 1. 用紙の大きさは、日本工業規格B列5番とすること。
2. ※印の欄は、記入しないこと。

様式第二

第 号	第 種情報処理技術者試験合格証書
昭和 年 月 日	本籍 (氏名)
通商産業大臣	年 月 日生
印	情報処理技術者試験規則第五条の規定によりこの合格証書を交付する

- 備考 用紙の大きさは、日本工業規格A列3番とすること。



付録 2 情報処理関係の標準・規格

目 次

I	標準化の現状と考え方（電子計算機および情報処理関係の標準化のための 体系調査報告書 — 日本情報処理開発センター資料から抜粋）	250
II	ソフトウェア標準化に関する調査 （日本電子工業振興協会ソフトウェア技術委員会標準化分科会調査報告 抜粋）	271
III	情報処理用流れ図記号 J I S C 6 2 7 0 - 1 9 7 0	317
IV	電子計算機プログラム用言語 F O R T R A N（水準 3 0 0 0） J I S C 6 2 0 3 - 1 9 6 7 解説（抜粋）	341
V	電子計算機プログラム用言語 C O B O L J I S 案 - 1 9 7 1（抜粋）	349

I 標準化の現状と考え方（電子計算機および情報
処理関係の標準化のための体系調査報告書＝日
本情報処理開発センター資料から抜粋）

目 次

1. 標準化の現状.....	251
2. 標準化の考え方.....	264

1 標準化の現状

1.1 ISOの標準化活動

工業標準化についての国際機関の1つとして国際標準化機構(International Organization for Standardization: ISO)がある。

ISOは戦前に組織された万国規格統一協会(ISA)の事業を引継ぎ1947年に結成され、現在55か国が加入している。

わが国は、閣議決定に基づいて、日本工業標準調査会が昭和27年に加入した。国際標準化事業は、関係各国の利害を話し合いの形で調整して、国際的に統一した規格を作り、各国がその実施の促進を図ることによって、国際間の通商を容易にするとともに、科学・経済等諸般の部門における国際協力の推進を目的としている。

国際標準化活動への参加は、わが国の主張を国際規格に反映させたり、国際的な規格制定の動向を掌握したりするなど、技術情報を収集することによって、国際的視野のもとにJIS(日本工業規格)を制定して、その国際性を高め、海外市場の拡大に資するためにきわめて重要である。

開放経済体制の進展に伴って、国際標準化事業の必要性はますます増大してきているが、加入各国はこれら事業のもつ重要性をよく認識し、その参加活動はきわめて積極的であり、これら会議に多数の代表を送り、自国の主張を反映させるように努力している。

現在ISOには、131の専門委員会があり、このもとに分科会や作業グループが設けられている。この中で、電子計算機と情報処理を担当しているのはTC97で、その組織図は図1のとおりである。

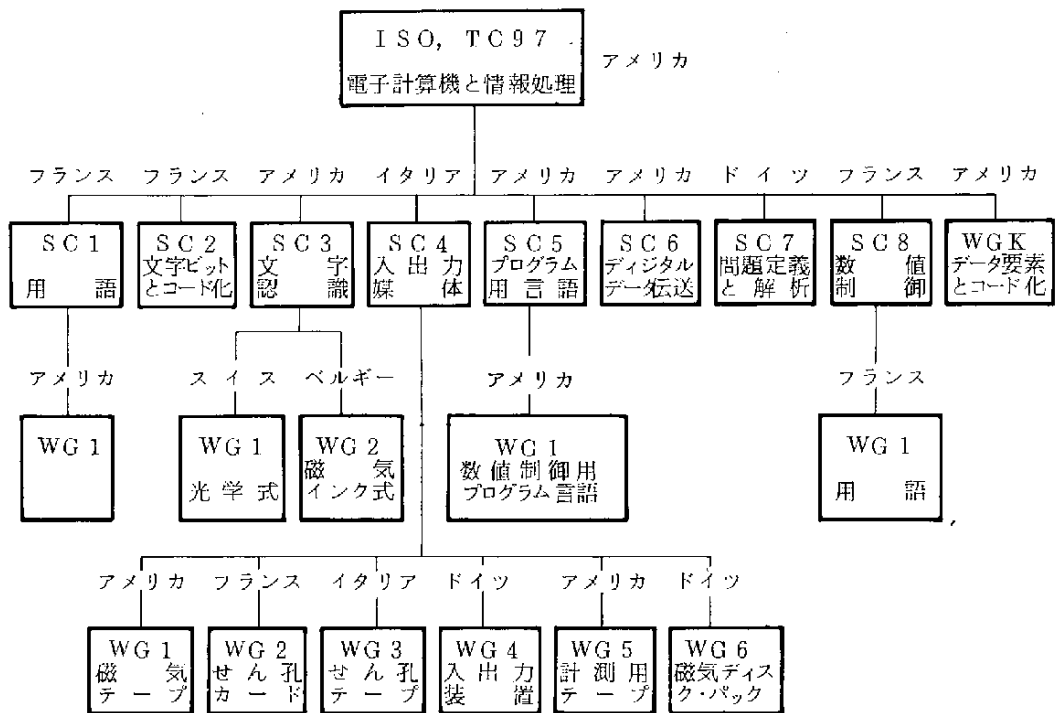


図1 ISO, TC 97 (国際標準化機構, 第97技術委員会) 組織図

注(1) 表中添記の国名は各グループの幹事国

(2) SCは分科会, WGは作業グループ

(3) 1971年現在の組織図を示す。

専門委員会には, さらに専門的な事項を調査, 審議するため必要に応じて分科会 (Subcommittee : SC) や作業委員会 (Working Group : WG) が設けられる。

各会員団体は既設専門委員会に自由に参加でき, これら3つの委員会に参加する場合の資格は

○積極的に参加して投票権を行使するPメンバー (Participation Member)

○資料の提供を受け, 会議に出席できるが, 投票権のないOメンバー (Observer Member)

○会議に出席せず, かつ資料の提供も受けないNメンバー (Non Member)

に分れる。

新しいISO資格案を作成するに際して, 専門委員会の幹事国は, 各会員団体からの資料および他の方面からの情報等を基にしてISO原案 (ISO Draft Proposal) 第1号を作成し, その要旨および説明書等とともに, 専門委員会を構成するPメンバー, OメンバーやIS,

O中央事務局に送付する。原案第1号に賛成が得られない場合は、幹事国は原案第2号を作成して、Pメンバーの意見を求める。このようにして原案がPメンバーによって実質的に承認されるまで、同様な過程が繰返される。

専門委員会の幹事国は、専門委員会によって承認された原案を、その審議過程等を記載した報告書とともに、ISO中央事務局に送付する。中央事務局は、この原案をISO推薦規格案(Draft ISO Recommendation)として全会員団体に送って、承認と意見を求める。この推薦規格案について、大多数の同意を得られない場合は、さらに第2案を作成して同様に承認を求める。

このような手続きは、Pメンバーの大多数および会員全体の60%以上の賛成投票が得られるまで繰返される。

Pメンバーの大多数および会員全体の60%以上の賛成により、最終的に承認されたISO推薦規格案は、理事会で審議し承認された後はじめてISO推薦規格となる。

さらに理事会は、この推薦規格をISO規格(ISO Standard)として採用すべきかどうかについても検討する。

1.2 JISの制定状況

わが国では原則としてISOで推薦規格または推薦規格案になったものを基本として、それに日本の実情を考慮してJISを制定することになっているが、現時点でのISO推薦規格、推薦規格案ならびに文書審議案とJISとの対照表は、表1に示すとおりである。表2は、今後のJIS化検討項目をまとめたものである。ISOの審議項目となっているが審議計画が未定のもの、42年に日本電子工業振興協会が実施した「電子計算機および情報処理法のJIS化のための体系調査報告書」および44年に日本情報処理開発センターが作成した「データ・コード標準化体系調査報告書」を基にした。

表1 ISO推薦規格、同推薦規格案、同文書審議案とJISとの対照表

*1 進行状況の略号(ISOの審議状況を示し、左側は1970.10.1現在、右側は1970.12.31の予定を示す。)

A=項目調査の段階

B=原案(Draft Proposal)作成段階

C=原案がPメンバーの過半数以上に承認された段階

D=ISO推薦規格案(Draft ISO Recommendation)が全加盟国に送付された段階

E=Draft ISO Recommendation がISO加入国の60%以上の賛成投票およびPメンバーの過半数により承認された段階

F=Draft ISO Recommendation がISO理事会に送付された段階

G=Draft ISO Recommendation がISO理事会によってISO Recommendationとして承認された段階

1 用語(SC1)

ISO推薦規格(R), 推薦規格案(DR) および文書審議案(N)	JIS規格, JIS規格案および原案作成
*進捗略号	
01:一般用語(97/1, N140) BC	(1) 情報処理用語C6230-7010
02:数字および論理(97/1, N151) AB	
04:データの構成(97/1, N101) BC	
05:データの表現(97/1, N105) BC	
06:データの準備および取扱い (97/1, N103) BC	
07:算数および論理オペレーション (97/1, N147) AA	
09:プログラムの形式化と準備 (97/1, N775) AA	
11:インストラクション (97/1, N144) -A	
情報理論用語(97/1, N146) AA	
ハードウェア(準備中) --	
コーディングおよびチェック (準備中) --	
データ通信(準備中) --	

2 キャラクタ・セットとコーディング(SC2)

(1) 情報交換用6および7ビット・キャラクタ	(1) 情報交換用符号C6220-69.6(R646)
-------------------------	-----------------------------

・セット (R646)	R963)
(2) 7トラック磁気テープ上での6および7ビット・キャラクタ・セットの表現 (R961)	
DG	
(3) 9トラック磁気テープ上での7ビット・キャラクタ・セットの表現 (R962)	(2) 情報交換用符号の磁気テープ上での表現 C6222-69.6 (R962)
DG	
(4) 7ビット・キャラクタ・セットから得られる4ビット・キャラクタ・セットの定義に対するガイド (R963)	(3) C6220に含まれている
DG	
(5) 磁気テープのラベルとファイル構成 (R1001)	(4) 情報交換用磁気テープのラベルとファイル構成 C6245-70.4 (R1001)
DG	
(6) せん孔テープ上での6および7ビット・キャラクタ・セットの表現 (R1113)	(5) 情報交換用符号の紙テープ上での表現 C6221-69.6 (R1113)
EG	
(7) 12欄せん孔カード上における7ビット・キャラクタ・セットの表現 (R1679)	
BG	
(8) 12欄せん孔カード上における7ビット・パターンの表現 (DR2021)	(6) 情報交換用符号の紙カード上での表現 4.5年度原案作成中 (DR1679, DR2021)
AD	
(9) MICRおよびOCR用キャラクタ・セットのコーディング (DR2033)	
CD	
(10) 7ビット・キャラクタ・セットとともに用いるコード拡張の手続き (DR2022)	
BB	
(11) バイナリ・データの記録方法 (97/2, N310, 273)	
AA	
(12) 情報交換用ディスク媒体のコード化 (97/2, N322)	
AA	
(13) エッジせん孔カードのコード化 (97/2, N323)	
(14) パックド・ニューメリック形式による10進数の表現 (97/2, N347, 354, 442)	
CB	

注) 1970年中にさらに調査を行なう	
(15) 7ビット・コードのコントロール・キャラクターの図的表現(DR2047)	C D
(16) 見出しラベル(97/2, N352)	A A
(17) 8ビット・コードのコード拡張とその構成(97/2, N399)	B B
(18) 3キャラクター・エスケープ・シーケンスの階級(97/2, N400)	

3 文字認識(S C 3)

(1) 磁気インク文字認識用字形および印刷仕様(R1004)	F G	(1) 磁気インク文字認識用字形および印刷仕様C6251-71.3(予定)(R1004)
(2) 光学文字認識用字形(英数字および記号)(R1073)	F G	(2) 光学式文字認識用のための字形(英数字および記号)C6250-70.6(R1073)
(3) 光学文字認識用印刷仕様(DR1831)	C D	(3) 光学文字認識用字形(カナ文字)45年度調査研究中
		注) この調査研究には、光学文字認識用印刷仕様(DR1831)も含まれる

4 入出力(S C 4)

(1) 情報交換用せん孔テープの寸法(R1154)	D G	(1) 情報交換用紙テープの孔の位置と寸法45年度審議予定(R1154)
(2) 未せん孔のカード仕様(DR1681)	D F	
(3) 80欄せん孔カードの角形せん孔の位置と寸法(DR1682)	D D	(2) 未せん孔紙カード, 45年度原案作成中(DR1681, 1682)
(4) 未せん孔紙テープの特性(DR1729)	D D	
(5) 計測用磁気テープのための一般用リールとハブ(DR1858)	D D	(3) 情報交換用紙テープC6243-70.4(DR1729)
(6) 計測用未記録磁気テープ(一般寸法)(DR1859)	D D	
(7) 計測磁気テープ用精密リール		

(DR1860)	DD	
(8) 情報交換用7トラック200RPI磁気テープ(DR1861)	BD	
(9) 情報交換用9トラック200RPI磁気テープ(DR1862)	BD	(4) 情報交換用磁気テープの情報記録様式, 45年度原案作成中(DR1862, DR1863, 97/4/1, N184)
(10) 情報交換用9トラック800RPI磁気テープ(DR1863)	BD	
(11) 情報交換用200, 800RPI, NRZ Iおよび1600RPI位相エンコードの未 記録磁気テープ(DR1864)	BD	(5) 情報交換用磁気テープ C6240-704(DR1864)
(12) 情報交換用9トラック1600RPI磁気 テープ(97/4/1, N233)	BB	(6) 情報交換用磁気テープ・リールC6242審 議中(DR1864)
(13) せん孔カードの用語 (97/4/2, N82)	AB	
(14) 特殊目的用カード (97/4/2, N74)	AA	
(15) せん孔テープについてのデータ交換のため の一般的要求(97/4/3, N129)	BB	(7) けん盤配列44, 45年度原案作成中
(16) 情報交換用1吋(25.4mm)せん孔テープ のためのリールおよびコア(97/4/3, N136)	BB	
(17) システム・フィロソファクタ (97/4/4, N13)	AA	
(18) I/O インタフェースの機能的必要条件表 (97/4/4, N63, 66)	AA	
(19) I/O インタフェース(接合条件) (97/4/4, N59, 60)	AA	
(20) 機械的互換性のための6枚磁気ディスク・ パックの物理的特性		

(97/4/6, N32, 43)	BB
② 磁氣的互換性のための6枚磁気ディスク・ パックの特性(97/4/6, N28,33)	AB

5 プログラム用言語(SC5)

(1) プログラム用言語ALGOL (DR1538)	DF	(1) 電子計算機プログラム用言語FORTRAN AN(水準7000)C6201-67.5 (DR1539)
(2) プログラム用言語FORTRAN (DR1539)	DF	(2) 電子計算機プログラム用言語FORTRAN AN(水準5000)C6202-67.5 (DR1539)
(3) ISO6および7ビット・キャラクタ・セ ットによるALGOLのためのハードウェア 表現(DR1672)	DD	
(4) プログラム用言語COBOL (DR1989)	DD	(3) 電子計算機プログラム用言語FORTRAN AN(水準3000)C6203-67.5 (DR1539)
(5) プログラム言語の標準化に適用すべき基準 (97/5, N221)	CC	(4) 電子計算機プログラム用言語ALGOL (水準7000)C6210, 67.5(DR1538)
(6) 機械の数値制御に用いられるプログラム用 言語(APTのようなもの)	BB	(5) 電子計算機プログラム用言語ALGOL (水準6000)C6211-67.5(DR1538)
		(6) 電子計算機プログラム用言語ALGOL (水準5000)C6212-67.5(DR1538)
		(7) 電子計算機プログラム用言語ALGOL (水準4000)C6213-67.5(DR1538)
		(8) 電子計算機プログラム用言語ALGOL (水準3000)C6214-67.5
		(9) 電子計算機プログラム用言語ALGOLの 入出力(水準70)C6215-67.5 (DR1538)
		(10) 電子計算機プログラム用言語ALGOLの

	入出力(水準60) C6216-67.5
	(1) 電子計算機プログラム用言語ALGOLの 入出力(水準50) C6217-67.5
	(2) 電子計算機プログラム用言語ALGOLの 入出力(水準40) C6218-67.5 (DR1538)
	(3) 電子計算機プログラム用言語COBOL 45年度原案作成中(DR1989)
	(5) 電子計算機プログラム用言語COBOL 45年度原案作成中(DR1989)

6 デジタル・データ伝送(SC6)

(1) 情報メッセージ中の誤り検出用水平パリティの用法(R1155) DG	(1) 伝送回線上のキャラクタ構成と水平パリティの用法C6360-71.4(予定) (R1155, R1177)
(2) スタート・ストップおよび同期伝達のためのキャラクタ構成(R1177) DG	
(3) データ通信システムのための基本的モードコントロールの手法(DR1745) DF	
(4) CCITT(国際電信電話諮問委員会)推薦規格V24が適用できるデータ端末機器とデータ通信装置間の回路交換用コネクタ・ピン・ナンバー(DR2110) CC	(2) 通信線のMODEMと通信制御装置のインタフェースC6361-71.4(予定) (DR2110)
(5) 基本的モード・コントロール手法の変更と拡張(97/6, N330) AB	
(6) コードに独立な伝送制御の手法(97/6, N328) AC	
(7) One-Way およびTwo-Way 伝送制御の手法(97/6, N325, 329) AA	
(8) 同期比(97/6, N248, 224, 326) AA	

(9) システム・パフォーマンス (97/6, N157)	AA
(10) 信号品質(97/6, N332)	AA
(11) データ・シグナリング・シート	AA
(12) 用語	AA
(13) リターン・チャネル	AA
(14) アコースティック・カップリング (97/6, N225, 313, 292)	AA

7 問題定義と解析(SC7)

(1) 情報処理のためのフローチャート記号 (R1028)	DG	(1) 情報処理用流れ図記号 C6270-70.4 (R1028, 97/7, N56)
(2) フローチャートの利用方法(97/7, N56)	BB	(2) 論理回路記法, 45年度審議予定
		(3) 電子計算機の性能表示, 45年度原案作成 中

8 数値制御(SC8)

(1) 数値制御用コード(R840)	DG	(1) 数値制御機械の座標軸と運動の記号 (審議中-R841)
(2) 数値制御機械の座標軸と運動の記号 (R841)	DG	(2) 数値制御用コードおよび紙テープ・フォー マット, 45年度原案作成中(R340, R1056, R1057, R1058, R1059)
(3) 数値制御機械用せん孔テープ・ブロック・ フォーマット(R1056)	FG	
(4) 数値制御機械用互換性せん孔テープ可変ブ ロック・フォーマット(R1057)	FG	
(5) 数値制御機械用せん孔テープ可変ブロック ・フォーマット(R1059)	FG	
(6) 数値制御用せん孔テープ固定ブロック・フ ォーマット(R1059)	FG	
(7) 数値制御機械の位置決めのためのせん孔テ		

ープ可変ブロック・フォーマット (97/8, N150)	BB
(8) 一般用語, プログラミング(97/8/1, N154)	BC
(注) SC1に移管	
(9) エラー, アキュラシーなどの定義 (97/8/1, N19)	AA
(10) 第3章から第6章まで (97/8/1, N33)	AA
(11) 検出端(トランスデューサ) (97/8, N133)	-A
(12) 数値制御機械用記号 (97/8, N151)	-B

9 データ・コード(WGK)

(1) 年月日の表現(DR2014)	DD	(1) 日付けの表示(コード)C6262-70.4 (DR2014)
(2) 週の番号付け(DR2015)	DD	(2) 時刻の表示(コード)C6263-70.4
(3) インタナショナル・システムにおける単位 の名称(SIユニット)の表現 (97/N314)	BB	(3) 都道府県コードC6260-70.4
		(4) 市区町村コードC6261-70.4
		(5) 性別コード(審議中)
		(6) 産業(業種)コード, 45年度審議予定
		(7) 商品コード, 45年度原案作成中
		(8) 勘定科目コード, 45年度原案作成中
		(9) 計量単位コード, 45年度原案作成中 (97/N314)
		(10) 職業コード, 45年度原案作成中

合 計	合 計
I S O 推 薦 規 格 R = 20 件	J I S 制 定 済 み 33 件 (R = 11, D R = 7)
I S O 推 薦 規 格 案 D R = 22 件	工 業 標 準 調 査 会 で 審 議 中, ま た は 審 議 予 定 2 件 (R = 1)
そ の 他 審 議 案 N = 48 件	45 年 度 原 案 委 託 10 件 (R = 8, D R = 4)
	45 年 度 調 査 研 究 委 託 1 件 (D R = 1)

表2 その他の情報処理関係JIS化検討項目

1 文字認識
(1) 磁気文字認識 (MICR) および光学文字認識 (OCR) のための信号レベル
(2) 文字認識用特殊文字 (漢字など) の字形化
(3) OCR および MICR 関係用語
(4) OCR 用手書き字形 (英数字, 記号, カナ)
2 入出力関係
(1) 磁気カードの仕様と記録方法
(2) チャネルと装置制御間のインタフェース (接合条件)
(3) 装置制御と入出力機器間のインタフェース, オンライン・タイプライタ, ディスプレイ, 紙テープ装置, 紙カード装置, 磁気テープ装置, 磁気ディスク装置, その他の入出力機器
3 プログラム用言語
(1) プログラム用言語 PL/I
(2) タイム・シェアリング用コマンド
(3) シミュレーション用言語
(4) システム性能評価基準 (GIBSON MIX に類するもの)
4 データ・コード関係
国名, 住居表示, 血液型, 続柄, 学歴, 職種, 職階, 職能, 技能, 官公庁, 学校, 研究機関, 企業 (会社名), 銀行, 金融機関, 事業所, 病院, 企業形態, 輸送機種, 船種, 固定資産品目,

取引決済条件，材料，土地，建築物など

注：上記項目は，下記の資料を基とした。

- (1) I S Oの項目となっているが審議計画未定のもの
- (2) 「電子計算機および情報処理方法のJ I S 化のための体系調査報告書」（日本電子工業振興協会調査，昭和42年）
- (3) 「データ・コード標準化体系調査報告書」（日本情報処理開発センター調査，昭和44年）

2 標準化の考え方

2.1 電子計算機と情報処理関係の標準化に関する考え方

2.1.1 標準化の必要性

電子計算機と情報処理に関する産業は、他の産業では類例のない勢いで成長の過程にある。成長のごく初期の段階では、技術的に若く、変遷の途上にあること、およびこの産業の持つ本質的な多様性のために、標準化の動きは、2～3の例外を除いてはあまり盛んではなかった。

しかし近年になって、このままこの産業が伸長し、多様化がさらに進んで行くと、製造者にとっても利用者にとっても不都合な点が増大して行くことが明らかになってきた。このため数年前から、世界的に標準化の動きが盛んになってきており、わが国もまた例外ではなくなった。

この産業の先進国グループと目される米、英、仏、独等の各国は、国内での標準化の検討を進めているが、他方、この標準化は世界的視野に立って行なう必要があることを痛感し、それぞれの立場から世界的規模の標準化の努力が要請されるようになった。

2.1.2 標準化項目の選定

標準化項目の選定に当っては、

- (a) その項目を標準化することが妥当であること。
- (b) その項目の標準化が要望されていること。
- (c) その項目の標準化が技術的に可能であること。
- (d) 標準化したら守らせることができること。

の4つの条件が必要である。

2.1.3 標準化の基本的態度

- (a) 世界の動向に注目して、できるだけ同一内容であることが望ましい。

昨今までは国内における外国機種シェアが強く、これらの機種に注目してきたため、おのずとある範囲については標準らしいものが存在してきた。今日必要な標準化は、外国の特定機種に注目することではなく、世界各国の標準化の動き、すなわちANSI、BSI、DNA、AFN、OR等で取上げられつつある項目、内容および国際的な動き、すなわちISO、ECMAの審議項目、内容について注目することを意味する。

- (b) わが国固有の問題を標準化するに当っては、従来の慣習にとらわれず、世界の動向と矛盾し

ないように心掛けることが望ましい。

わが国固有の問題はデータ・コードを除けば、主として国字の取扱いとの関連が大部分を占める。この関係は前述の特定の外国機種を基本として何とか国字の取扱いを行なえるようにする傾向が強かった。この標準化に当っては、情報処理関係者だけの中で考えないで、利用者はもとより広く各界の有識者に見解を聞き、これと情報処理技術との調整を計るようにすることが望ましい。

この結果が世界の動向と矛盾しないようにするためには、ISOとの連絡を密にし、ISOの場にわが国固有の問題を反映するよう努力することも怠ってはならない。

(c) 標準化に当っては、技術の進歩の妨げにならないように留意すること、適切な時期を見て改訂することなどが望ましい。

(d) 標準化する時期の設定を誤らないことが必要である。標準化が早すぎると、技術のパラエティを制限することになり、遅すぎると、実績に押されて標準化を困難にするばかりでなく、標準化したものが有名無実になることもある。特に進歩途上の技術については、標準の時期は熟慮断行が必要である。

(e) 標準化の順序を誤らないようにすること。比較的相互に関連が薄い項目は、標準化順序を考慮しなくても矛盾を生ずることは少ないが、密接な関連がある項目は、同時に標準化することも含めて、順序を間違えないようにしないと、因果関係のために標準化しにくくなる場合があるので、注意を要する。

2.2 データ・コードJIS化に当たりの問題点

標準化の考え方は上述のとおりであるが、データ・コードのJIS化については、データの性格から特有の問題点があるので、注意が必要である。

2.2.1 ISOにおけるデータ・コード

アメリカはdata-elementとそのコード表示に互換性を持たせようと1965年にISOにその作業を提案した。これに対してヨーロッパ諸国は、日時、単位、数値などの表示のような基本的なelementについてさえ各国は少しずつ異なっていて、万国共通の標準がない現状であるためデータ・コードの標準化作業に対してはきわめて消極的であった。

その結果、1970年の会議でアメリカは現実的に考えて、対象を日付、単位のような最小限度の基本的なものに限ろうと申し出て、ようやく認められるに至った。

2.2.2 データ・コードのJIS化の困難性を出发点

標準化は場合によって業界、ユーザ団体等比較的小さいグループの中でも行なえるが、これが国家的規模になると、その対象とする範囲がかなり限定されてくる。特にデータ・コードは、国民全体に関連するものが多いからといって、どの業界やグループにも共通的に適用できるものにするためのJ I S化を考えると、現実的には限られた項目にならざるを得ないだろう。

従来、製造規格などのJ I S化は、特定の業界やその他の限られたグループを中心に進められてきたが、データ・コードの場合は、特定の業界ばかりでなく、業界、業種、さらに地域を越えた非常に広範囲な利用を考慮しなければならない。

したがって、そのニーズが多様多岐にわたるため各業界やユーザ・グループ間の調整が非常に困難な問題となってくる。すなわちデータ・コードの対象となるエレメント1つにしても、それぞれ独自の見方、考え方をしているため当然①分類の方法、②粗密の程度、③配列の順序、④表記の方法などで要求がかなり違ったものとなってくる。

このようにデータ・コードのJ I S化には他の場合と異って、特殊かつ困難な事情があることを銘記しなければならない。

このようなことから、現在の段階で将来のデータ・コードのJ I S化の方策を明確に打出すことはきわめてむずかしい。

これとともに現在の段階では、

- ① データ・ベースの形式がまだ発展段階にあること。
- ② データ・コードと自然語との関連の研究が不足していること。
- ③ 情報処理技術の進歩に見合うだけのデータ・コードの研究がされていないこと。
- ④ 電子計算機の利用領域の拡大の見通しが確立されていないこと、などの情報処理全般にかかわる問題もからんで、いっそう問題を複雑にしている。

仮に商品名コードを設定するとして、次の点について、要点を考えてみよう。

- (1) 範囲：これが大きくなると、けた数がふえる。電話番号を例にとれば、東京での番号は7けたであるが日本全体で話す際は(03)を加えなければならず、外国ではさらに(081)を加えなければならない。
- (2) 専門化：仮に魚を例にとれば、魚の種類ははなはだ多く、学者がつける魚の名前と業者のつける名前とは必ずしも一致しない。商品として価格を念頭に入れば、魚の目方(重量)、鮮度、季節などいくつかのパラメータを含めねばならない。ところが業界ごとに必要なパラメータは違うし、また違ってもさしつかえない。したがって、適用業務を専門ごとに決めることで初めて商品名コードを論ずることができる。このような業界限りのコードは使用範囲も当然業界内に限ら

れ、すなわち業界内の標準化とJIS制定とは別個の問題であるので、業界内のコードは、それぞれ業界中で自発的に標準化すべきものと考えられる。

(3) データの形式(Format) : 自動車の番号を例にとれば、次のようなものである。

品川51

ゆ

9456

この中で(品川)(51)(ゆ)(9456)のそれぞれは、data-elementと見られ、それぞれ意味を持っている。漢字もあれば平仮名もある。このようにコードを制定する作業では、数字(番号)化に限定すべきでなく、文字または文字と数字との併用を考慮する必要がある、それと同時にdata-formatを決めることも重要なことである。

また次の例として年月日をとる。昭和46年2月17日を710217と表わせばデータ・コードである。このコードは年月日にそれぞれ2けたずつ与えてこの順に配置させ、計6数字で表示するというFormatを決めたことになるのである。

つまりdataとdata-formatとdata-elementの関係が、どうすれば合理的であり、矛盾のないものとなるのか、その方法を学識と洞察をもって議論を重ね、秩序を立てて次第に明確なものを求めて行くこと、および現にわかっていて、間違いのないことを中心とした指導要領を開発することなどが、データ・コードJIS化の出発点といってよいだろう。

2.2.3 データ・コードのJIS化へのユーザの要請

データのコード化は、ユーザの例からみれば現在の技術ではComputer内部での処理、および媒体上での表現(特にOCR, MICRなど)の効果を高めるために必要である。前述のようにデータ・コードのJIS化については問題点と困難性が多々あるが、そのJIS化は情報交換の円滑化をはかる上から、可能な限り進められることが望ましいとする要請も強い。

JIS化についてのユーザの要請のうち重要と思われるものをあげると、次のとおりである。

(a) 1970年代にはComputer Readable Language 媒体の市販が予想されるので、まず企業環境に関する企業外データに盛込まれるべき内容のコード統一を優先すべきである。企業環境に関する統計データは政府官庁、もしくは情報提供産業が提供することになるだろうから、まず官庁統計から実施すべきである。

(b) data-code-elementの個々の意味づけをするとcodeけた数が多くなるので、単なるアイデンティフィケーションだけでよいのではないか。元来data-code-elementに意味づけを行なうのはPCSの名残りではないかと思う。

(c) 企業は業界内で標準化した独自のコードをすでに使っている場合が多いので、個々の企業に J I S 化コードの使用を強制することは避けてもらいたい。

したがって、本委員会は以上の諸点を考慮し、データ・コード標準化の問題を使用する立場から、

- ① 情報の流通性の増大、すなわち情報交換の円滑化
- ② 情報の共同利用の促進、すなわち情報の社会的共同化
- ③ 情報の全国的掌握とその利用のための全国共通データ・コード体系の採用

などを考慮して、これらの要求に適合するような普遍的コードについて全国的あるいは国際的に、統一化をはかることが必要であろうと考える。

2.2.4 J I S 化の対象と範囲

データ・コードの標準化または J I S 化に当たっての基本的な態度は上述したとおりであるが、J I S 化を進めるに当たってその対象と範囲と優先順位とを具体的に定めるためには、3つの基準、すなわち J I S 化の必要性、緊急性、実現性について総合的に判断する必要がある。

第1の必要性については、

- ① データ・コード体系の全体を全国的・一義的に統一化する必要のあるもの。
- ② データ・コード体系の全体を全国的に一義的統一をすることが困難な場合、そのコードの一部の共通部分だけでも全国的に調整することが望ましいもの。
- ③ 全国的に調整すべき共通部分の標準化も困難な場合、相互交換の基準を確立しておくべきもの。

第2の緊急度の評価の基準としては、次の4つがあげられる。

- ㉚ 全国的な情報交換の実施のため、データ・コードの J I S 化が緊急であるかどうか、すなわち、行政的活動や産業界における全国的組織活動等について、情報交換を全国的に必要とするような場合に、最も緊急性があると考えられる。
- ㉛ 情報を全国的共通な基盤のうえから掌握して行くために、データ・コードの J I S 化が緊急性のあるもの、すなわち、行政機関や産業機関が意思決定する手段として、全国的に統計資料を掌握し、それを利用するためには全国共通のコード体系を採用することを必要とする場合である。
- ㉜ 各分野で部分的標準化が実現されつつあるが、将来さらに広い地域、範囲および水準にわたって、標準化を必要とする時期が到来することが予想され、標準化方針の確立が遅れると、不都合をきたす恐れのある場合である。

㊤ 対象が比較的明確で、J I S 化の実施が容易であり、情報交換、共同利用等に際して便益が比較的得やすい場合である。

第3の実現性は、前項㊤は最も容易であり、J I S 化の効果も期待できる。㊦、㊧は必ずしもJ I S 化は容易な仕事ではないが、必要性、可能性のいずれも高いものと認められる。㊨については軽視できないが、各分野間の利害の調整が困難であるだけに、その実現性は低いと考える。

Ⅱ ソフトウェア標準化に関する調査

日本電子工業振興協会ソフトウェア技術委員会
標準化分科会調査報告（昭45.3） 抜粋

目 次

1	まとめ	272
2	データの標準化	275
3	プログラミング言語の標準化	286
4	ソフトウェア技術文書の標準化	301
5	プログラム作成作業の標準化	303

1 ま と め

標準分科会は、ソフトウェア全般について標準化の動向、問題点を検討した。

昭和44年度の調査にあたっては、昭和43年度の調査項目を含め、さらにソフトウェア外注の増加、ハードウェア・ソフトウェアの価格分離等の動向から、ソフトウェア生産上の問題として、作業、評価、道具等の項目を追加した。前半は主としてすでにISOあるいはJISとして検討が進められているもの、後半に追加したものは、現段階では必ずしもISO、JIS等で取上げられてはいないが、ソフトウェアの生産あるいは運用の面から、何らかの規準、標準が望まれているものという形になっていると考える。

以下各章別に概要を述べる。

データ関係

情報交換用符号については、すでにISO勧告およびJIS制定が行なわれており、今後はその実行上の問題を検討することになるが、ISOにおいては次の段階として、7ビット・コードの拡張が審議されている。わが国のJISは、ISO 7ビット・コードを基として、8単位符号も制定しており、一部ながらISOに先行した形となった。欧米各国に比較して、カナ、漢字等必要とする文字数の多いわが国の要求は、同様に7ビット・コードの拡張を必要とする各国の代表として、国際標準の中に十分盛込むよう今後の検討の要があろう。

データの媒体上での表現としては、情報交換用符号の磁気テープ、紙テープ上の表現法がJIS化され、磁気テープのラベルとファイル構成についても、近く正式制定の予定である。磁気ディスクについては、直ちにJIS化を行なうのは無理としても、今後の重要な研究課題であろう。

また、データ・コードについても都道府県コードほか数件が近くJIS化されるのに続いて、つぎつぎと検討が進められており、SE活動や各ユーザー会等を通して得られた広いユーザーの要望とともに、メーカーとしての意見もJISに反映されるよう、調査が必要であろう。

言語関係

プログラミング言語では、先にJIS化されたFORTRAN、ALGOLに続いて、COBOLのJIS化が取上げられているが、後述のように国際的な標準案にも、なお流動的な点があり、その動向に十分な注意が必要であろう。

以上の一般的なコンパイラ言語のほか、問題向き言語、グラフィック言語、会話型言語などを

調査しているが、いずれもその適用分野自体が研究開発途上のため、現在標準化を行なうのは時期尚早の感があり、今後も国内、国外の動向を調査してゆく必要がある。

表記法関係

表記法のうち、流れ図については近く J I S (C 6 2 7 0) として制定予定であるが、ソフトウェアの文書を中心として、その他の項目は統一的な標準制定は非常にむずかしい。これらの標準化は各方面から要望され、特に大規模のシステム開発には不可欠とされ、假々に何らかの解決策を図っている現状であるが、具体的な形で公表されているものはきわめてまれである。本報告ではその中の一例として、工業技術院の大型プロジェクトに適用されている書類規定を中心として、表記法の標準化についての考察を述べた。この方面の動向については今後も広く資料を集め、調査を継続する必要がある。

用語については、前回の結論に基づきその実施方法を検討の結果、現在工技院の大型プロジェクトと電電公社の D I P S 両計画において、J I S 化を考慮に入れつつ用語の標準化を行なっているので、当分科会はその編集、保守を行なっている日本ソフトウェア(株)と連絡を保ちながら標準化を進めることとした。

作業の標準化

新システムを開発したり、あるいは既存のシステムの維持、改善を図りながら運営してゆくには、その生産性向上の見地から標準化は重要な課題である。このような標準は、全システムに適用し得る工業規格、あるいは国際会議で審議される標準とはその性格が若干異り、近い将来に制定することはほとんど不可能であろう。しかし、ソフトウェア生産の作業進行管理等の上で、何らかのよりどころが求められている現況から、ここでは参考例を示すこととした。今後はこれらの中で、部分的に標準化可能なものを検討し、その積上げによって逐次改良を図ってゆく必要がある。

情報処理技術者の呼び方と仕事の記述の標準化

前項の作業の標準化に関連して、ソフトウェア生産の組織の構成単位である情報処理技術者の呼称と、その任務の内容が問題となるが、最近 I S O / T C 9 7 / S C 1 で検討を開始されることになった。現在習慣的に使われているこれらの名称・内容が必ずしも統一されておらず、したがって社会的な評価、あるいは要員養成等の面から、十分な対策が立ち遅れているとも考えられる。すなわち、この標準化に当っては、単なる現状の整理、統一のみでなく、情報処理技術者をどうあらしめるべきかを十分考慮すべきであろう。

ソフトウェアの評価

ソフトウェアの評価が問題とされる場面としては、設計指針としての性能の目標、コストに係る作成期間・工数等の見積り、完成したものに対する検査基準、修正の難易も含めた運用に当っての使いやすさの比較、あるいは、取引の対象として考える場合には、その価格決定の基準等、広範囲の場合が考えられる。また評価する者の立場によっては、同一のソフトウェアに対して異った評価が下されることもありえよう。

したがって、評価に関して発表された論文等も多いが、普遍的な評価基準、あるいはその標準化は、現状においては至難の業と考えられる。しかし、ソフトウェアの比重が高まり、商品としての取扱いが要求されるようになっている状況から、ソフトウェアの評価方法を確立し、標準化を推進することは必要であり、たとえば、一部の科学技術計算等のアプリケーション・プログラムに対しては、その適用範囲、制度、制限条件等を明確にすれば可能と考えられるものもあり、そのようなものから逐次進めてゆくことが必要であろう。

そのための施策として、設計目標やドキュメンテーションの確立、評価項目の細分化・定量化等を図り、各種のデータを収集し、あるいはシミュレーションやベンチマーク等評価のために有効な技法の開発を推進することが必要であろう。

標準化の道具

ソフトウェア生産に当って、その道具となるものの標準化として、今回は流れ図とシステム記述用語について調査結果を報告する。これらは、ソフトウェア流通分野の拡大および自動作成化に伴って、標準化の必要性が高まっている。

流れ図については、前述のように近く J I S として制定の予定であるが、流れ図の源である問題の論理構成、プログラム機能、さらには表現の精粗にも個人差があり、J I S 制定後も直ちに十分な効果を常に発揮するとは限らない。また、その用法については、I S O の Recommendation が発行寸前で各国のクレームによって再審議されることになり、わが国は J I S が先行することになるので、両者間に矛盾を生ぜしめないようにしなければならない。

システム記述用語は、開発用語の項で述べるように、オブジェクト・プログラムの効率向上とのバランスが問題であるが、新システム開発のロスを少なくし、プログラムの標準化を目指すものとして、今後も調査研究が必要である。

2 データの標準化

2.1 情報交換用符号とその拡張

(1) 情報交換用符号

数年にわたって審議されてきた情報交換用符号が昭和44年6月1日に日本工業規格JIS C 6220として制定された。国際的にはすでに国際標準化機構(ISO)がISO R 646として国際標準を定めている。今回のJISは、このISO R 646を基本として、国内用として変更が許されている部分等をわが国の事情に合わせ、また、カナ文字についても同時に規格を制定したものである。

この規格の適用範囲は、情報処理およびデータ伝送を行なうシステムの情報交換に用いる符号で、装置内部の符号は規定されないが、オンラインのシステム等異なるシステム間の情報交換が次第に増加している現状から、データ伝送の分野から次第に新規格への切替えが行なわれよう。今は現用機械の増設、更新、手持ちファイル量等から切替えの要否、時期、方法をどうするかが問題となろう。また、今回は時期尚早として取上げられなかった漢字その他の文字、キャラクタについても、後述のように符号の拡張としてISOで検討が開始されているので、新技術による周辺機器、端末機器の制御キャラクタ等とともにその動向に注意する必要がある。

(2) 7単位符号の拡張(Code Extension)

ISO R 646は、6単位および7単位の符号について標準を定め、その拡張の手段としてESC, SO, SIおよびDLEの4個のキャラクタを使用するようになっている。その具体的な使用法は、ISO/TC97/SC2(文字セットとコーディング)で検討中であるが、標準7単位符号(以下ISO-7と略記)から同じく7単位符号系に拡張するものをCode Extension、8単位符号系に拡張するものをCode Expansionと呼んでいる。現在はいずれも伝送制御キャラクタを除き、残りの制御キャラクタと、記号、数字、文字を表わすグラフィック・キャラクタの追加に限られており、特殊機能キャラクタである拡張(ESC)、シフトイン(SI)、シフトアウト(SO)を使用する。

(a) ESCシーケンス

ESCには次に続く1符号の意味を変える前置符号としてのほかに、他の符号系に移ることを意味する場合もある。そこで、一般的にESCと、その後続くいくつかの符号とを1組にして、

符号系を変える補助制御機能と考え、ESCシーケンスと呼ぶ。すなわち、ESCシーケンスは (ESC)(I)⋯(I)(F) の形で表わされ、最初は常にESCとし、最後(Final)のキャラクタ(F)との間に中間(Internal)のキャラクタ(I)が必要に応じてそう入される。ESCとFのみのシーケンスは2キャラクタESCシーケンス、Iが1個入るのは3キャラクタESCシーケンスと呼ばれる。4キャラクタ以上のシーケンスも考えられるが、現在の案では国際標準は3キャラクタまでを対象としている。

ここで、(I)はISO-7表の第2列16個のキャラクタの中から選び、(F)は第3列～第7列のキャラクタ(ただし、抹消(DEL)を除く)79個の中から選ぶことになっている。

さらに、ESCシーケンスについて国際標準として、登録制をとることになり、専門家グループがその登録方法をどのようにするかを検討、立案した。

(b) 2キャラクタESCシーケンス

ESC(F)の型式のもので、後述のように単独の制御キャラクタを追加指定するのに用いられる。Fはこの場合、次のように3群に分けられる。

0	1	2	3	4	5	6	7
			Fa	Fe		Fa	
			Fp			Ep	

ESC(Fe)⋯Feは第4、第5列の32キャラクタの1つで、このシーケンスは、8単位符号表の標準が定められたとき、その第08、09列にそれぞれ配列されるべき制御キャラクタを指定する。たとえば、ESC Gは8単位符号表の第08列、第7行に配置されるべき制御キャラクタを示す。

ESC(Fa)⋯追加される制御キャラクタのうち、国際標準とされるものを示す。

ESC(Fp)⋯追加される制御キャラクタのうち、国際標準から除外して、Private またはdo it yourselfとして扱われるものを示す。

FaとFpは、第3列、第6列または第7列のキャラクタで、その合計は47となるが、それぞれをいくつずつにするかはまだ定められていない。

(c) 3キャラクタESCシーケンス

ESC (I) (F) の型式のもので、単独またはグループ制御キャラクタ、グラフィックのキャラクタ群の追加、および ISO-7 表中の国内用とされている部分について何種類かを作った場合、それらを選択、指定するのに用いる。

(I) はこのシーケンスの機能を分類、指定するのに用い、ISO-7 表の第2列16キャラクタの中から指定される。1969年5月ナポリ会議の後で設けられたSC2の専門家グループにより、現在下記のように8キャラクタについて提案されているが、第2列の残りのキャラクタも将来Iを拡張するため保留されている。

(F) は、原則としてIによって分類された機能を、特定のキャラクタ(群)に適用するためのもので、ISO-7 表の第3～第7列のキャラクタ79個の中から選択される。

現在提案されている3キャラクタESCシーケンスは次のとおりである。

128キャラクタを定義するもの

ESC * (F) … ISO-7 表の中で、別のキャラクタで表現することが許されている部分のみを置き換えるほかに、ISO-7 表と全く同じ符号表を指定する。このシーケンスによっていったんある表が指定されたなら、SOを用いなくてその表が適用される。別の表に移るときはFの異なるESC * (F) シーケンスによる。

ESC ^ (F) … 以下の条件を満足することによって、ISO-7 表と互換性(Compatible)のある128キャラクタの符号表を指定する。

- 第0列および第1列は制御キャラクタのみであること。
- 伝送制御、NUL、SO、SI、CAN、SUB、ESC、DELの各キャラクタはそのままとして変更しないこと。
- 第2列～第7列はグラフィックのみであること(DELは除く)。
- ISO-7 表と同一のグラフィックをそう入する場合は、その位置を変更しないこと。

ESC & (F) … 上記2項のいずれにも該当しない128キャラクタの符号表を指定する。このシーケンスで指定する符号表に、制御キャラクタとしてISO-7 表と異なる意味を与える、7単位符号で正常な機能を果たす機器がうまく動作しない恐れがあり、また、標準符号に戻る事が不可能になることもありうる。

符号表の一部を定義するもの

ESC % (F) … 第2～第7列のグラフィックについて、別の符号表を指定するもの。この符号表の中のキャラクタを使用する場合はSOを併用する。もし、送受両者間で、SOで使用する符号表がただ1つと定められている場合は、必ずしもこのシーケンスを使用しなくても

よい。しかし、当事者間に同意が得られていない場合や、S Oで使用する符号表が1種類より多い場合には、特定の符号表を指定するために、このシーケンスが必要である。

ESC + (F) …第0列および第1列の制御キャラクタに追加するための、1組32キャラクタの制御キャラクタ表を指定する。この符号表の中の個々の制御キャラクタを使用するときは、7単位符号系では2キャラクタESCシーケンスを併用するものとし、そのシーケンスのFはISO-7表の第4列または第5列からとるものとする。

単独のキャラクタを定義するもの

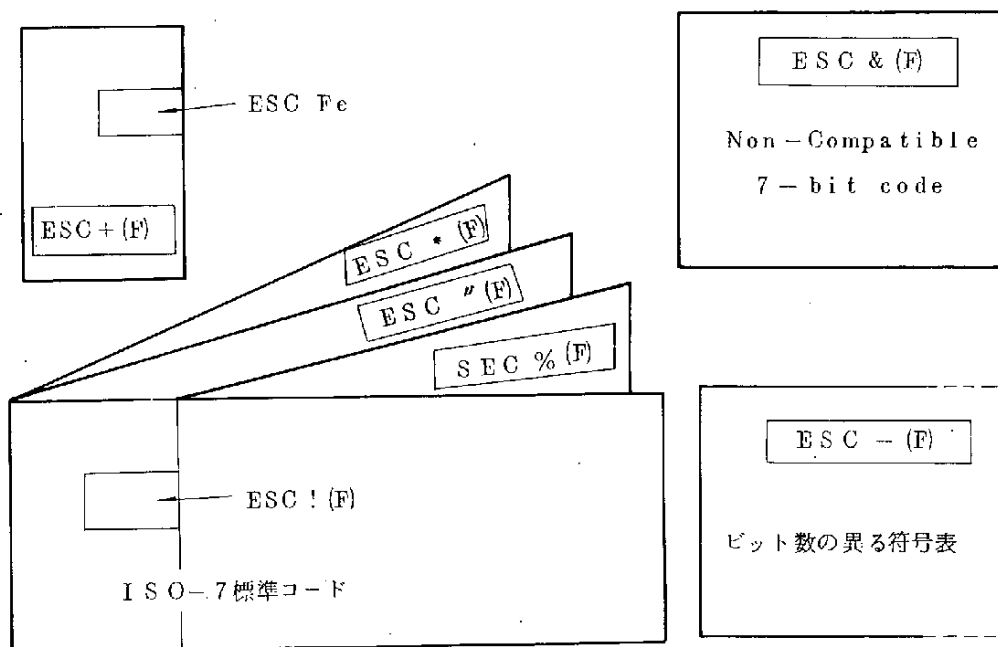
ESC ! (F) …単独の制御キャラクタを追加し、直ちにその機能を行なわせるもの。

その他

ESC - (F) …現在使用中の符号と異なる単位数の符号表を指定する。

ESC / (F) …上記のどれにも該当しないもののために保留する。ただし、“非該当”のクラスを設けるものではない。

以上のESCシーケンスの概念を図解すると、次図のようになる。



注) FeはISO-7表の第4, 第5列からのキャラクタであることを示す。(F)は前述のようにして最終キャラクタとして認められた79キャラクタ中の1個であることを示す。
枠なしのESCシーケンスは、単独の制御キャラクタであることを示し、枠で囲んだES

Cシーケンスは、セットになったキャラクタ群を示す。

(d) S O, S Iの用法

S O, S Iはグラフィックの拡張にのみ用い、制御キャラクタ等の拡張には適用されない。すなわち、32個の制御キャラクタ、DEL、およびESCシーケンス中に使用されるキャラクタの意味はS O, S Iの有無によって変更されない。

ESC%(F)シーケンスによっては、シフトの状態は変らない。そのため、1つの拡張された符号表から別の拡張された符号表に移るのは、いちいちS Iで標準セットに戻さなくともよい。

ESC%(F)とS O, S Iの使用例

a...ISO-7表の標準グラフィック・キャラクタ

b...ESC%Bシーケンスで指定された符号表(B)のグラフィック・キャラクタ

c...ESC%Cシーケンスで指定された符号表(C)のグラフィック・キャラクタ

q...ISO-7表に置き換えられるべき符号表として、ただ1つが当事者間で定められた場

合の符号表のグラフィック・キャラクタ

とすると、一般的にESC%(F)による指定とS Oによる変換は

$$\cdots a \overset{E}{S} \% B a \cdots a \overset{S}{O} b b \cdots b \overset{S}{I} a \cdots$$

または

$$\cdots a \overset{E}{S} \% B \overset{S}{O} b \cdots b \overset{S}{I} a \cdots a \overset{S}{O} b \cdots$$

符号表(B)から符号表(C)に変換するには

$$\cdots b \overset{E}{S} \% C c \cdots c \overset{S}{I} a \cdots$$

または

$$\cdots b \overset{E}{S} \% C \overset{S}{I} a \cdots a \overset{S}{O} c \cdots$$

変更する符号表がただ1つの場合は

$$\cdots a \overset{S}{O} q \cdots q \overset{S}{I} a \cdots$$

(3) 8単位符号表への拡張(Code Expansion)

ISO-7 表を拡張して、8 単位符号表の標準構成を定めるのには、いろいろあるが、最初取上げられたのは次の 2 案であった。

第 1 案：追加する 1 ビット位置を ISO-7 表の第 5 ビットと第 6 ビットの間とし、ISO-7 表と同じキャラクタは、新 8 単位符号表の第 6 ビットと第 8 ビットの値を 0 または 1 の同じ値とするもの。

第 2 案：追加する 1 ビットの位置を最上位の第 8 ビットとして、ISO-7 表と同じキャラクタは、この値が 0、追加するキャラクタは 1 となるようにするもの。

第 1 案は、NUL と DEL とがそれぞれすべて 0、すべて 1 の形となり自然であること、制御キャラクタの追加が必要なときは、32 種類までは ISO-7 の制御キャラクタに連続して拡張できることなど、理論的な特長があり、第 2 案は ISO-7 表のキャラクタの区別が簡単であるという特長がある。わが国は、カナ文字使用の上からも 8 単位符号の必要性が高く、早くからこの問題について検討を加え、1 案を支持していたが、文字の追加についてわが国ほどの必要性がなく、アルファベットの大文字と小文字の配列が離れることをきろう欧米各国によって第 2 案が採択されるに至った。

以下昨年（昭和 44 年）5 月にナポリで開催された ISO/TC 97/SC 2 の会議で検討された英国の提案をもとにして、現在の模様を述べる。

(a) 8 単位符号表の構成

① 16 行×16 列の配列で、列番号を 00 から 15 までとし、7 単位符号表の 0～7 と区別する。ただし、00～07 の内容は 0～7 と同じである。

第 8 ビット = 0								第 8 ビット = 1								
列 行	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	制 御	記 号	ISO-7 128 キャラクタ				英 小 文 字	英 大 文 字	制 御	追加する 128 キャラクタ						
1																
2																
⋮																
⋮																
⋮																
⋮																
⋮																
⋮																
⋮																
⋮																
⋮																
⋮																
⋮																
15																

② 08, 09の2列は伝送制御用を除く制御キャラクタの追加用とする。その試案として、英国からは次のような提案があった。

行 \ 列	0 8	0 9
0		
1		
2		
3		
4	CC 1 (文 字 抹 消)	
5	CC 2 (文 字 そ う 入)	DC 5
6	CC 3 (行 抹 消)	DC 6
7	CC 4 (行 そ う 入)	DC 7
8	CC 5 (B S)	DC 8
9	CC 6 (H T)	DC 9
10	CC 7 (L F)	DC 10
11	CC 8 (V T)	DC 11
12	CC 9 (ホ ー ム)	DC 12
13	CC 10 (N L)	DC 13
14	CC 11 (ス ペ ー ス)	DC 14
15	CC 12 (消 去)	DC 15

試案中のCCはCursor Controlの略で、ブラウン管表示のディスプレイを想定したものである。別の機器に対しては異った機能となるが、類似の機能を持たせるのが望ましいとしており、次のような意味としている。

CC 1 (Delete Character)

カーソルの右側の1文字を抹消し、かつその行の文字を左に1字分うめて空隙をなくす。

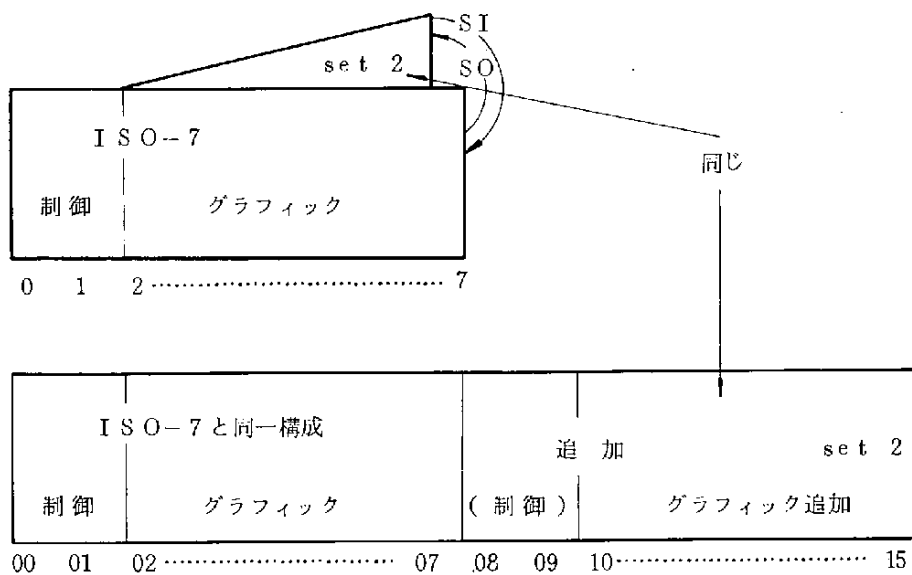
CC 2 (Insert Character)

カーソルの右側に1字そう入し、その行の後の文字を1字分ずつ右にずらす。

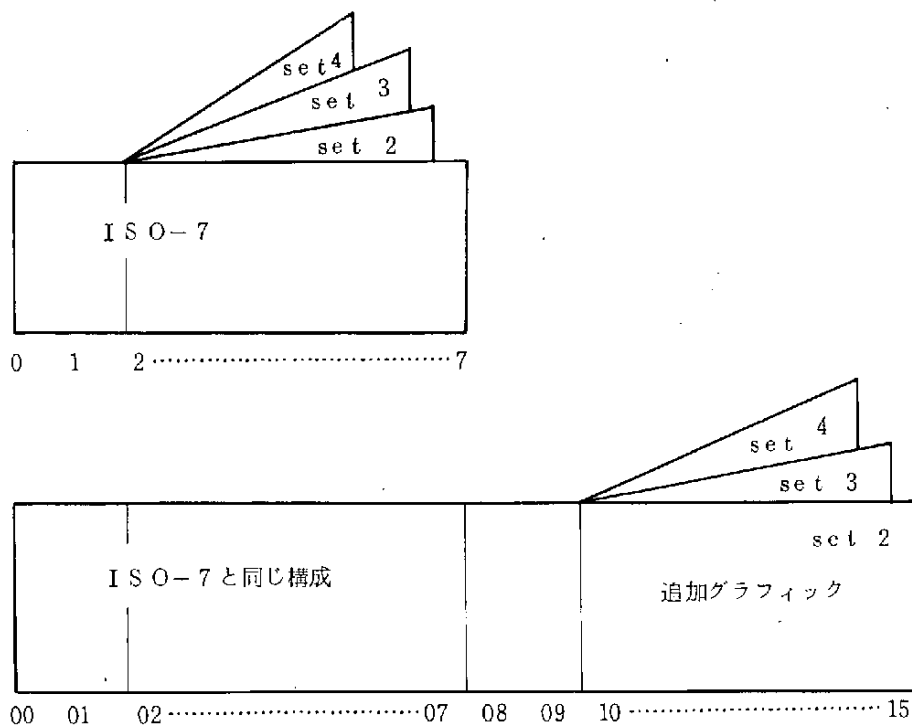
CC 3 (Delete Line)

1行抹消して、その下の行を上につめる。

CC 4 (Insert Line)



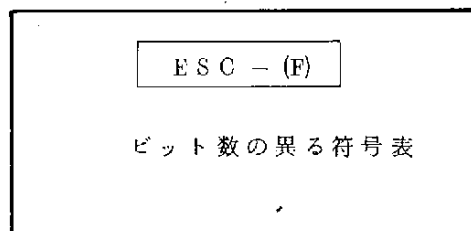
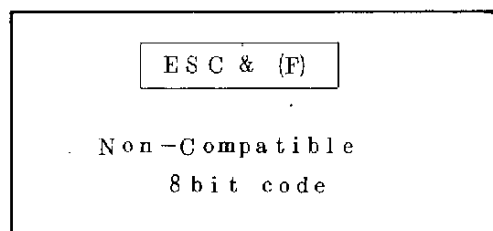
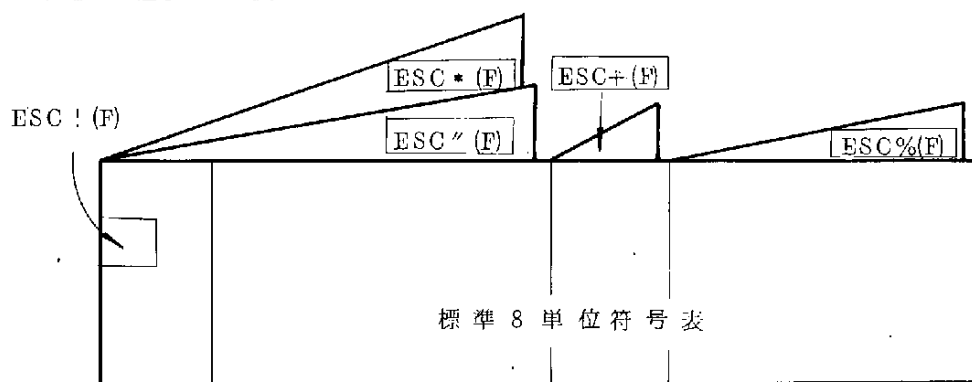
追加する符号表が2組以上ある場合は、



図のように8単位符号表の後半の部分に追加され、ESCシーケンスによって区別される。ただし、7単位の場合はSO、SIが併用されたが、8単位ではESCシーケンスのみで、SO、SIは不要である。

(c) 8単位符号表のESCシーケンス

概念的に図示すると次のようになる。



注) Feは標準ISO-7表の第4、第5列からのキャラクタを示す。(F)は最終キャラクタに認められた79キャラクタ中の1個を示す。枠で囲んだESCシーケンスは、セットになったキャラクタ群を示し、枠で囲んでいないESCシーケンスは、単独の制御キャラクタであることを示す。

ESC * (F) および ESC \" (F)

7単位の場合と同様の機能であるが、00～07の列に対してのみ有効とする。

ESC % (F)

10列から15列までに対して有効である。SOの使用は必要でないので、その使用は認められない。

ESC + (F)

7単位の場合はこのシーケンスとESC (Fe) シーケンスを併用することになっているが、

8単位の場合はESC(Fe)シーケンスは使用しない。

ESC&(F)

標準符号表とCompatibleでない符号表。ただし、標準とする符号表を前半、後半いずれの128キャラクタとするか、あるいは256キャラクタ全部にするか結論がでない。

以上のように情報交換用符号の国際標準は現在ISO R 646による7単位符号が中心となっているが、数学用の記号、天気情報交換のための記号、新開発の各種周辺、端末機器の制御用キャラクタ等の符号化の必要性が起っている。文字についても、欧米以外の国では英文字のみでは不十分なところが多い。わが国は片カナについてJISが制定されたばかりであるが、平かな漢字についてもその必要性、対象範囲、方法等について検討を加え、わが国の意見を早期に国際会議の場に反映させて、国際標準に折込む努力が必要である。

3 プログラミング言語の標準化

3.1 COBOL

(1) はじめに

COBOLは事務データ処理のプログラミング言語として、現在最もよく使用されている共通言語の一つである。COBOLはFORTRAN, ALGOLに比べて、次の2点により、特にその動向を注目しなければならない。特にCODASYLの動向には注意する必要がある。

(a) COBOLはCODASYL(COnference on DAta SYstem Language)によって仕様の決定が行なわれているが、常に新しい事務データ処理に適した言語にするため、仕様の改訂、追加がかなりはげしい。

(b) CODASYLによる仕様の決定とは別に、その範囲内で標準化の作業が進められている。ISOではすでに推薦規格案が出されて具体的な内容について審議中であるし、日本でも、JIS化の検討が始められようとしている。

昭和43年度の報告書では、COBOLについて、発展過程、標準化の歩み、JIS化の問題点等について、くわしく報告されているので、この報告書では、主に

- ・昭和44年(1969年)における、CODASYL, ISO, JIS, 国内メーカーの動向
- ・今後の方向と問題点

について報告し、次に

- ・昭和44年(1969年)に改訂されたCODASYL, COBOLの内容
- について概説する。

表1は、これまでの動向を、参考までに一覧表としてまとめたものである。

最後には、参考文献を示した。

表1 1959年～1968年の動向

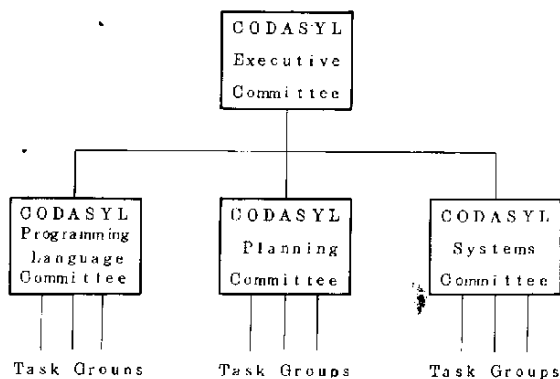
年	CODASYL	ISO	JIS
1959	5月 CODASYL発足		
1960	4月 COBOL-60の発行		
1961	6月 COBOL-61の発行		
1962			
1963	COBOL-61 Extended の発行		
1964			
1965	11月 COBOL1965年版の 発行		
1966			
1967		4月 ISO/TC97/SC5 (USA-25)176の発行 11月 パリでSC5の会議開催 上の資料を審議	
1968 (昭和 43年)	7月 COBOL 1968年版 (1969年9月までに7版まで 改訂されている)		JIS化の 方向にある

(2) 昭和44年(1969年)の動向

CODASYL, ISO, JIS, 国内外のメーカーにおける昭和44年(1969年)の動向は次のとおりである。

CODASYL

CODASYLの組織は1968年7月下図のようになった。



CODASYLはこれまでCOBOL-60, COBOL-61 Extended, COBOL Edition 1965を作成し、発足時に予定した大きな仕様追加は終了した。しかしその後の、いくつかの状況変化により、現在も作業は積極的に進められている。すなわち、

- COBOLの仕様決定を行なうのは、CODASYLがただ一つの機関であることが確認された。
 - USASI, ISO, ECMAなどもCODASYL COBOLのサブセットのかたちで標準案を作成しようとしている。
 - そのためCODASYLは仕様の明りょう化に努力せざるを得なくなった。
 - 通信回線やデータ・ベースなど、新しい事務データ処理としての使い方が生れ、COBOLとしても検討しなければならない。
- などの理由によるものと考えられる。

1968年にはCOBOL Edition 1965をベースにして、その後の改訂をすべて整理した“COBOL Journal of Development 1968”が発行された。これはCODASYLが承認している最新のCOBOL仕様書である。

昭和44年(1969年)9月までに7回改訂版が出されて、いくつかの新しい仕様追加が行なわれた。主な仕様改訂の内容については後述するが、大きな項目としては、次のようなものである。

- (a) 通信回線制御に関する機能の追加
- (b) EXAMINEの削除とINSPECTの追加

CODASYLにはProgramming Language Committee (プログラミング言語委員会)の下に次のTask Groups (小委員会)があって、主に新しい仕様の検討を行なっている。

- (a) Data Base
- (b) I/O Editing and Data Representation
- (c) Asynchronous
- (d) Report Writer
- (e) Communications

データ・ベース(Data Base)については仕様案が完成し、CODASYLの理事会で承認され、各方面に配布されている。1970年には、データ・ベースの仕様が正式仕様になるものと考えられるが、今後の重要性からみて、正式決定までには、かなりの議論を呼ぶものと考えられ

る。

入出力編集(I/O Editing and Data Representation)小委員会は前述のとおり、昭和44年(1969年)にEXAMINEを削除し、拡張されたINSPECT命令を正式に追加した。その他、データの取扱いに関する検討が続けられている。

非同期処理(Asynchronous)は、仕様の明確化について検討中である。

Report Writerは、すでに仕様が決定されているので、その使用経験から、詳細な追加仕様の検討が続けられている。

通信回線制御(Communications)は、前述のとおり、昭和44年(1969年)に新たに仕様追加が行なわれた。

ISO

昭和44年(1969年)には、特にめだった動きはない。1970年にはCOBOL勧告案が出されるものと予想される。

JIS

昭和44年(1969年)には、特にめだった動きはない。1970年から標準化の作業が開始される。

国内外のメーカー

米国ではUSASI COBOLが1968年に制定されたのでIBMなどを中心にUSASI COBOLの仕様に従ったコンパイラが作成され始めている。

国内では各種のコンパイラが作成されているが、ほとんどCOBOL Edition 1965に準じたものである。

(3) 今後の方向と問題点

昭和44年(1969年)はCODASYLによる積極的な改訂作業が進められたが、標準化に対する正式決定は何もなされなかった。

昭和45年(1970年)にはCODASYLの作業が更に進められると同時に、標準化作業が、いよいよ活動を開始する方向にある。すなわち、国際的には、ISO勧告が1970年内に配布されるであろう。これは1967年に出されたUSASI作成のISO/TC97/SC5 (USA-25) 176 [参考文献⑨]に1967年11月パリで開催されたSC5会議の審議結果、およびその後の審議結果が追加されたものとなるであろう。一方、国内では、工業技術院が1971年JIS制定を希望しているので、1970年には情報処理学会を中心に何らかの作業が開始されるはずである。

J I S 化をする場合、I S O 勧告案に準ずる形になるものと思われるが、次のような日本特有の問題もあり、J I S 制定の時期のみを急ぐのではなく、十分な審議をかきねるべきものと考えられる。

- (a) カタカナの扱い
- (b) 小型計算機におけるCOBOLの取扱い

CODASYLが将来の検討項目として取上げている項目は次のとおりである。

TOPIC GUIDELINE LIST (1969年5月)

- (1) Debugging Tools
- (2) Executive Interaction
- (3) Input / Output Editing
- (4) Library Maintenance
- (5) Extended Segmentation: Data and Procedure
- (6) Mathematical Functions
- (7) Floating Point
- (8) Default Options
- (9) Free (Reference) Form COBOL
- (10) Mass Storage Extensions
- (11) Shorthand COBOL--Abbreviations
- (12) Realignment
- (13) COBOL Meta-language
- (14) Generalized Subscripting
- (15) List Processing (Data Base)
- (16) Closed Subroutines
- (17) Collating Sequence
- (18) Global Common
- (19) Punctuation Characters (Use of)
- (20) Communications Facility
- (21) Variable Length Data Items
- (22) "ELSE" in Implied Conditional Statements
- (23) Index for COBOL Journal of Development

(24) Review of COBOL Glossary

(4) CODASYL COBOLの仕様

CODASYLで決定しているCOBOL仕様は、表1にも示したとおり、COBOL-60からCOBOL1968年版まで発行され、更に1969年(昭和44年)にも、改良、追加が行なわれた。以下に各COBOL仕様での主な追加内容と、1969年の変更内容について概説する。

COBOL-60

正式には"COBOL-A Report to the Conference on Data Systems Languages, including Initial Specifications for a Common Business Oriented Language (COBOL) for Programming Electronic Digital Computers"(COBOL-CODASYLへの報告書、デジタル電子計算機プログラミングのための事務用共通言語(COBOL)の第一回仕様書)という長い表題の文書で、以後COBOL60と呼ばれた。

COBOL-61

正式なCOBOL仕様書の第2版である。COBOL-60を1年間にわたる使用経験にもとづいて改訂したもので、COBOL-60と完全な共通性はない。このCOBOL-61は、ひろく利用され、現在作成されているほとんどのCOBOL翻訳ルーチンのもとになっている。以後発行されたCOBOL 1965年版や1968年版も、すべてCOBOL-61に対する改良、追加と考えられる。

COBOL-61 Extended

基本的にはCOBOL-61である。主な追加と変更は次のとおりである。

- (a) SORT(分類)機能の追加
- (b) Report Writer(報告書)機能の追加
- (c) ADD, SUBTRACT命令にCORRESPONDINGの追加
- (d) 各仕様に対する明りょう化

COBOL Edition 1965

この版はCOBOL-61 Extended と次の追加、変更から成る。

- (a) Mass Storage Files(大記憶ファイル)に対する読込み、書出し、処理のための一連の追加
- (b) IndexingとSearchの書き方を含むTable Handling(表操作)機能の追

加

- (c) 各仕様に対する明りょう化およびエラー診断表示の規約を削除

COBOL-Journal of Development-1968

COBOL Edition 1965をベースにして、その後の改訂をすべて整理したものである。主な改訂内容は次のとおりである。

- (a) プログラムの分割とその連絡の機能の追加
- (b) PICTUREと同様の句の削除(CLASS, POINT, SIZE, SIGNなど)
- (c) IDENTIFICATION DIVISION以外のすべてのDIVISIONでのCOPY使用句とINCLUDEの削除
- (d) ページ・オーバーフローの定義とテスト追加
- (e) Type 4の省略形は削除
- (f) DEFINEの削除
- (g) DIVIDEの余り, 追加
- (h) NOTEとREMARKSの削除(注記はつなぎ領域に*印を書く)
- (i) グラフィック・ディスプレイ用にSUSPENDの追加
- (j) 各仕様の明りょう化

更に1969年9月までに7回の改訂版が出されており、そのなかで次のような改訂が行なわれている。

- (k) 通信回線制御に関する機能の追加
- (l) EXAMINEの削除とINSPECTの追加

昭和44年(1969年)の主な改訂内容

昭和44年(1969年)末までに改訂された主な内容は次のとおりである。

- (a) 通信回線制御の機能
- ① DATA DIVISIONは次の5つのSECTIONから成る。
 - FILE
 - WORKING-STORAGE
 - LINKAGE
 - REPORT

COMMUNICATION

(CONSTANT SECTIONは削除された)

② COMMUNICATION SECTIONの書き方の形式は次のとおりである。

Format 1

```

CD cd-name; COPY library-name
[
  REPLACING { word-1
               { identifier-1 } BY { word-2
               { identifier-2 }
               { word-3
               { identifier-3 } BY { word-4
               { identifier-4 } } ]-----]

```

Format 2

```

CD cd-name; FOR INITIAL INPUT
[
  [ ( ; SYMBOLIC QUEUE IS data-name-1 )
    ( ; SYMBOLIC SUB-QUEUE-1 IS data-name-2 )
    ( ; SYMBOLIC SUB-QUEUE-2 IS data-name-3 )
    ( ; SYMBOLIC SUB-QUEUE-3 IS data-name-4 )
    ( ; MESSAGE DATE IS data-name-5 )
    ( ; MESSAGE TIME IS data-name-6 )
    ( ; SYMBOLIC SOURCE IS data-name-7 )
    ( ; TEXT LENGTH IS data-name-8 )
    ( ; END KEY IS data-name-9 )
    ( ; STATUS KEY IS data-name-10 )
    ( ; QUEUE DEPTH IS data-name-10 ) ]
  [ data-name-1, data-name-2, ..., data-name-11 ]
]

```

Format 3

```

CD cd-name; FOR OUTPUT
[ ( ; DESTINATION COUNT IS data-name-1 )
  ( ; TEXT LENGTH IS data-name-2 )
  ( ; STATUS KEY IS data-name-3 )
  [ DESTINATION TABLE OCCURS
    integer -2 TIMES
    ( ; INDEXED BY index-name-1 [ , index-name-2 ] ... ]
  ( ; ERROR KEY IS data-name-4 )
  ( ; SYMBOLIC DESTINATION IS data-name-5 )
]

```

③ PROCEDURE DIVISIONの書き方は次のとおりである。

C-1 MCS (Message Control System)とMessage のやりとりを記述する命令として次のものが追加された。

$$\begin{array}{l}
 \underline{\text{RECEIVE}} \text{ cd-name } \left\{ \begin{array}{l} \underline{\text{MESSAGE}} \\ \underline{\text{SEGMENT}} \end{array} \right\} \underline{\text{INTO}} \text{ identifier-1} \\
 \quad [; \underline{\text{NO DATA}} \text{ imperative-statement}] \\
 \underline{\text{SEND}} \text{ cd-name } \underline{\text{FROM}} \text{ identifier-1} \\
 \\
 \underline{\text{SEND}} \text{ cd-name } (\underline{\text{FROM}} \text{ identifier-1}) \left\{ \begin{array}{l} \text{WITH identifier-2} \\ \text{WITH } \underline{\text{ESI}} \\ \text{WITH } \underline{\text{EMI}} \\ \text{WITH } \underline{\text{ETI}} \end{array} \right\} \\
 \\
 \left[\left\{ \begin{array}{l} \underline{\text{BEFORE}} \\ \underline{\text{AFTER}} \end{array} \right\} \text{ ADVANCING } \left\{ \begin{array}{l} \text{identifier-3 LINES} \\ \text{integer LINES} \\ \text{PAGE} \\ \text{mnemonic-name} \end{array} \right\} \right]
 \end{array}$$

C-2 MCSと端末機器との間の回線の制御をする命令として次のものがある。

$$\begin{array}{l}
 \underline{\text{ENABLE}} \left\{ \begin{array}{l} \underline{\text{INPUT}} \quad (\underline{\text{TERMINAL}}) \\ \underline{\text{OUTPUT}} \end{array} \right\} \text{ cd-name ;} \\
 \text{WITH } \underline{\text{KEY}} \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\} \\
 \underline{\text{DISABLE}} \left\{ \begin{array}{l} \underline{\text{INPUT}} (\underline{\text{TERMINAL}}) \\ \underline{\text{OUTPUT}} \end{array} \right\} \text{ cd-name ;} \\
 \text{WITH } \underline{\text{KEY}} \left\{ \begin{array}{l} \text{identifier-1} \\ \text{literal-1} \end{array} \right\}
 \end{array}$$

C-3 通信機能だけで利用されるとは限らないが、この分野での利用価値の大きな命令として、文の切離しや結合をする次のものがある。

STRING { identifier-1 } , [identifier-2]
 { literal-1 } , { literal-2 }
 { identifier-3 }
 DELIMITED BY { literal-3 }
 { SIZE }
 [{ identifier-4 } , [identifier-5]
 { literal-4 } , { literal-5 }]
 DELIMITED BY { identifier-6 }
 { literal-6 } } -----
 { SIZE }
 INTO identifier-7 [; WITH (POINTER identifier-8)
 [; ON OVERFLOW imperative-statement]
 UNSTRING identifier-1
 [DELIMITED BY [ALL] { identifier-2 }
 { literal-1 }
 [, OR [ALL] { identifier-3 }
 { literal-2 }]]
 INTO identifier-4 [, DELIMITER IN identifier-5]
 [, COUNT IN identifier-6]
 [, identifier-7 [, DELIMITER IN identifier-8]
 [, COUNT IN identifier-9]]
 [; WITH POINTER identifier-10]
 [; TALLYING IN identifier-11]
 [; ON OVERFLOW imperative-statement]

(b) EXAMINE命令が削除され、機能を拡大したINSPECT命令が新たに追加された。
 形式は次のとおりである。

INSPECT identifier-1
 TALLYING { , identifier-2 FOR { , { { ALL } { literal-1 }
 { LEADING } { identifier-3 } }
 CHARACTERS }
 { { BEFORE } INITIAL { literal-2 } }
 { AFTER } { identifier-4 } } } }

INSPECT identifier-1

$$\begin{aligned} & \text{REPLACING} \left\{ \left\{ \begin{array}{c} \text{ALL} \\ \text{LEADING} \\ \text{FIRST} \end{array} \right\} \left\{ \begin{array}{c} \text{literal-3} \\ \text{identifier-5} \end{array} \right\} \text{BY} \left\{ \begin{array}{c} \text{literal-4} \\ \text{identifier-6} \end{array} \right\} \right. \\ & \left. \left[\left\{ \begin{array}{c} \text{BEFORE} \\ \text{AFTER} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{c} \text{literal-5} \\ \text{identifier-7} \end{array} \right\} \right] \right\} \dots \end{aligned}$$

INSPECT identifier-1

$$\begin{aligned} & \text{TALLYING} \left\{ \text{identifier-2} \text{ FOR} \left\{ \left\{ \begin{array}{c} \text{ALL} \\ \text{LEADING} \\ \text{CHARACTERS} \end{array} \right\} \left\{ \begin{array}{c} \text{literal-1} \\ \text{identifier-3} \end{array} \right\} \right\} \right. \\ & \left. \left[\left\{ \begin{array}{c} \text{BEFORE} \\ \text{AFTER} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{c} \text{literal-2} \\ \text{identifier-4} \end{array} \right\} \right] \right\} \dots \end{aligned}$$

$$\begin{aligned} & \text{REPLACING} \left\{ \left\{ \begin{array}{c} \text{ALL} \\ \text{LEADING} \\ \text{FIRST} \end{array} \right\} \left\{ \begin{array}{c} \text{literal-3} \\ \text{identifier-5} \end{array} \right\} \text{BY} \left\{ \begin{array}{c} \text{literal-4} \\ \text{identifier-6} \end{array} \right\} \right. \\ & \left. \left[\left\{ \begin{array}{c} \text{BEFORE} \\ \text{AFTER} \end{array} \right\} \text{INITIAL} \left\{ \begin{array}{c} \text{literal-5} \\ \text{identifier-7} \end{array} \right\} \right] \right\} \dots \end{aligned}$$

(c) その他

① ACCEPT命令で、DATE、DAY、TIMEからデータを受取ることができるようになった。

② CALL命令に記憶装置のオーバーフローを検出する機能が追加された。

(d) Procedure Divisionの動詞は次のとおりとなる。

Category	Verbs
Arithmetic	ADD
	COMPUTE
	DIVIDE
	INSPECT(TALLYING)
	MULTIPLY
	SUBTRACT

Asynchronous Processing	{ HOLD PROCESS
Compiler Directing	{ COPY ENTER USE
Conditional	{ ADD(SIZE ERROR) COMPUTE(SIZE ERROR) DIVIDE(SIZE ERROR) IF MULTIPLY(SIZE ERROR) READ RETURN(END) SEARCH SUBTRACT(SIZE ERROR) WRITE(INVALID KEY or END-OF-PAGE)
Data Movement	{ INSPECT(REPLACING) MOVE STRING UNSTRING
Ending	STOP
Input-Output	{ ACCEPT CLOSE DISABLE DISPLAY ENABLE OPEN READ RECEIVE SEEK SEND STOP(literal) SUSPEND WRITE

Inter-Program Communicating	{ CALL CANCEL
Procedure Branching	{ ALTER CALL EXIT GO TO PERFORM
Report Writing	{ GENERATE INITIATE TERMINATE
Sorting	{ RELEASE RETURN SORT
Table Handling	{ SEARCH SET

(5) 参考文献

(a) CODASYL COBOL関係の資料

① CODASYL; "COBOL-A Report to the Conference on Data Systems Languages, including Initial Specification for a Common Business Oriented Language (COBOL) for Programming Electronic Digital Computers"; DEPARTMENT OF DEFENSE(1960-4)

② CODASYL; "COBOL-Report to CONFERENCE on DATA SYSTEMS LANGUAGES Including REVISED SPECIFICATIONS for a COMMON BUSINESS ORIENTED LANGUAGE (COBOL) for Programming Electronic Digital Computers"; DEPARTMENT OF DEFENSE(1961-6)

③ CODASYL; "COBOL-61 Extended"; DEPARTMENT OF DEFENSE(1963)

④ CODASYL; "COBOL EDITION 1965"; DEPARTMENT

OF DEFENSE(1965)

⑤ CODASYL; "COBOL, Journal of Development-1968"
(1968)

注) ①, ②, ③, ④がそれぞれCOBOL-60, COBOL-61, COBOL-61
Extended, COBOL-65の正式な仕様書である。

⑤はCODASYLが承認している最新の仕様書である。1969年9月までに7回改訂版が
出されている。

(b) 日本語訳

⑥ "和訳COBOL"; 情報処理学会(1963-3)

⑦ 関根智明監訳; "COBOL1965年版"; 情報処理学会(1968-9)

注) ⑥は上記②の日本語訳である。⑦は上記④の日本語訳であるが、1968年1月までの変
更と改訂を完全に含み、1968年7月までの変更と改訂も明示してある。

(c) ISO関係の資料

⑧ USASI; "pUSASI COBOL Standard, CIB#9, ISO/TC97
/SC5(USA-25)176"(1967-4)

(d) その他

⑨ 西村恕彦; "コボルの現状"; 情報処理学会

第11回プログラミング・シンポジウム報告集(1970-1)

3.2 FORTRAN

FORTRANは科学技術計算用のプログラミング言語として、最も普及している共通言語の
1つである。各言語のなかでも、コンパイラ開発、普及が一番はやかったことから、標準化の作
業も、最初に行なわれた。

現在、ISO推薦規格案(1965年)、JIS規格(1967年5月1日制定)が決ってい
る。

昨年度の報告書で"FORTRAN発展の歴史", "日本工業規格制定にいたる経過"につい
て詳しく報告されているので、今回は昭和44年の動向についてのみ報告する。

昭和44年、FORTRANに関する標準化の作業は、ISO、JISともに、特にめだった
動きはない。しかし、昨年度の報告書でも指摘されているとおり、現在のJIS規格にはいくつ
かの問題点があり、それらについての検討は各方面で続けられている。JISは3年ごとに改訂

作業をしてよいことになっているので、昭和45年(1970年)には、具体的な作業が作業グループによって開始される予定である。

改訂の際に考慮すべき項目としては、

- ① 現JIS規格を使用して、実用面で遭遇した問題に対する項目
- ② 現JIS規格の発展、充実に関する項目

に分けられる。

前者は主に、条文の説明が前後の章で調和していないもの、規定をより厳密にした方がよいもの、現規定ではプログラムの書き方に不都合を生ずるものなどである。これらは、昨年度の報告書と同様、現規格に沿って作成されたFORTRAN処理系を包含するような形で改善されるという条件のもとで、できるだけ速かに処理されることが望ましいと考えられる。

後者は、いくつかの問題点を含んでおり、賛否両論がある。ここでは以下に項目の羅列にとどめるがJIS規格への適用については慎重な態度で臨むべきものとする。

- (1) ハードウェアの進歩、計算機の使用形態の発展に応じて機能を拡張していくのは当然としても、技術計算のFORTRAN(またはALGOL)と事務用のCOBOLという2系統の言語が独立に機能拡張をしていくのは望ましいことか。
- (2) 機能拡大をするとき、現規格に含めるべきか、オプションにすべきか。
- (3) FORTRANはオペレーティング・システムの開発に先行したため、ハードウェア的色彩が強いが、一貫思想のもとによくまとめられた言語である。オペレーティング・システムのことを考慮した新しい機能を追加していくことが、本当にFORTRANを使いよくしていく方向であるか。
- (4) JISの3000, 5000, 7000の水準について再検討が必要であるという意見がある。更に高水準のものを作ること、および最近の動向からみてMINI-COMPUTER用の更に低水準の仕様の検討が必要であると考えられる。
- (5) オンライン応用のためのステートメントの検討
- (6) TSS用FORTRANの標準化の検討
- (7) その他ダイレクト・アクセス装置に関するステートメントの追加や、ビット・ストリング操作の規定など比較的詳細な検討項目については、昭和43年度の報告書で報告されているので参照してほしい。

4 ソフトウェア技術文書の標準化

(1) 背景

ソフトウェアの生産は、工場制手工業あるいはもっと極端に鍛冶屋的であるといわれる。ソフトウェアの生産の手法がこれまで確立できなかった最も大きな理由として、研究開発的な色彩が非常に強く管理が困難であることがあげられている。それとともに比較的小きなシステムの場合は、鍛冶屋の手法で徒弟制度的な人事管理によって作成するのがむしろ能率的であると考えられていたためであろう。

近年、作成するシステムが大きく複雑になるにつれて従来のやり方では非能率であることがわかり、さらにそのような手法では実際にソフトウェアが完成し得ないということが認識はじめられ、いろいろな面での標準化が推進されている。ここに記述するドキュメントの表記法の標準化は、その最も大きな課題の1つで、最近活発に議論されている。しかし、昭和45年にJISが制定された流れ図（フローチャート）を除いて、他のものは各企業体ごとに標準化の努力がなされているのが現状で、このような状態がまだ1、2年は続くと思われる。しかし表記法の標準化は、直接生産性につながる重要な問題であることが認識されているので近い将来、国内あるいは国際的規模で推進されることは間違いないであろう。

(2) 標準化の対象にする文書

ソフトウェアを作成、使用、あるいは維持してゆく過程で作成される文書類には、次のようなものがあげられる。

- (a) 計画やねらいを記述したもの
- (b) 構成や機能を記述したもの
- (c) 内部方式を記述したもの
- (d) 流れ図やリスティング
- (e) 検査方法を記述したもの
- (f) 保守について記述したもの
- (g) 利用者向けの説明書

表記法の標準化は、これらの文書類の作成と維持について、記述する項目とその詳しさに対する水準、表現手段、管理体系などについての標準を決めることである。

(3) 標準化による効果

表記法を標準化することの効果は、生産管理の面と人事管理の面とで議論されるべきであろう。特に前者については、標準化された文書が効果的に管理されるとすぐにでもソフトウェアの原価低減が図れるように考えられているようであるが、この考え方には疑問を感じる。従来、ソフトウェアの価値は、あたかも磁気テープやカードなどの媒体に入れられたものだけであるようにみて、作成された文書類の量や質を問わない傾向にある。文書類もソフトウェアの1つであるから、その価値は正しく評価されなければならない。事実文書類を標準化することによって、ソフトウェアそのものの標準化を促し、変更が容易になり、有能なアナリストをプログラミングにくぎづけにする弊も避け得るといった広汎な利益が生れるのである。したがって生産性は上に述べた広義のソフトウェアについて評価されるべきであって、そういう意味で標準化によって生産性が向上すると考えられる。

(4) 表記法の例

表記法が1つのプロジェクトの中で標準化された例として、通産省工業技術院の大型プロジェクトで開発するソフトウェア・システムに関するものを別紙にあげる。

この文書は、

- ソフトウェアに関する公式書類規定
- ソフトウェアに関する公式書類規定に対する解説

の2つの部分から成っている。この種のもは、各方面の企業体で作られていると考えられ、それぞれが実施されて、成果や問題点が報告されることが望まれる。

5 プログラム作成作業の標準化

プログラム作成作業とデータ処理の仕方を全体的に標準化していくことは、開発の生産性を向上させる見地から、今後重要な課題になると思われる。しかし、すべてのシステムに同一の規準を適用することは不可能なことである。もし可能だったとしても、望ましいことではないであろう。したがって現行のシステムの事情を十分考慮しながら、それと矛盾しないような独自の規準を決めていくことが無理のない方法ということになる。

プログラム作成作業の標準化を検討することは、必要に応じて独自の規準を決める際の際のよりどころを与えるものさしを検討することでもある。

規準が決ると、その規準にあった方法にどう切替えてゆくかということが重要になる。全く新しいコンピュータやプログラミング・システムを採用する場合には、はじめから規準に従って作業を進めることができるが、現行のやり方を少しずつ変えながら規準の適用範囲を広げてゆく場合は、混乱を起さないように配慮しなければならない。

5.1 規準の必要性

- (1) コンピュータが現在のように広く使われるようになると、他の部門と同様、プログラム作成やデータ処理についても、水準の高い原価計算や品質管理を行なわなければならない。
- (2) 新しいコンピュータを設置したり、よりレベルの高い新しいシステムを開発しようとするとき、既存のプログラムが一定の標準的な方法にもとづいて書かれていれば、新しいシステムへの移行が容易にできる。
- (3) プログラムの内容がその人だけしかわからないような仕組みになっていると、プログラムの作成者がプログラムの保守についても、最後まで責任を持たなければならない。しかし、プログラム作成の作業が規準に沿って行なわれ、規定の方式で文書化されていれば、だれでもそれを解読できるし、プログラム作成後の変更を保守担当者にまかせることもできる。プログラマは、保守の責仕から解放されて新しい仕事にエネルギーを投入することができるのである。
- (4) 途中から作業に加わったプログラマは、それまでの作業の状況をは握しやすいし、標準的な作業の仕方だけを学べばよいので、早く生産的な仕事にとりかかることができる。
- (5) 標準的な方法が採用されていて、十分に文書化されていれば、管理者やコンピュータに関連

している部課のプログラマやオペレータが必要なときに容易に内容を理解できる。

5.2 標準化のための資料

(1) ISO (International Standards Organization), ANSI (American National Standards Institute, Inc. 旧称USASI), BSI (British Standards Institute) などを出しているプログラム言語, 流れ図記号などについての標準仕様

(2) JIS規格(日本工業規格)の仕様

プログラム言語について, FORTRAN言語とALGOL言語の規格がある。

流れ図記号, 磁気テープ・ラベルなどについて規格がある。

情報処理用語についての規格がある。

文書化については, 規格票の様式についての規格がある。

(3) SHARE, GUIDE, British Computer Society, ECMAなど, コンピュータ・ユーザの団体やメーカ団体がデータ処理についての規程を發表している。

(4) 情報処理学会, 電気学会, 通信学会などで情報処理用語その他の標準仕様を検討している。

(5) コンピュータ・メーカが各種の標準仕様書を出している。

(6) IBMから, “データ処理部門の組織化”, “データ処理規程書作成の手引き”の2冊の規程書がでている。

5.3 規程の徹底

規程は守らなければ何の意味もない。したがって守りやすい規程が現実的な良い規程ということになる。守りやすい規程にするためには, 必要以上に細かなところまで規定しないようにして, 水準を低めに抑えた規程にすることが重要である。また, 規程を作る場合には, 守りやすい規程にすると同時に, 確実に守らせるために, 検査体制を作ることが必要である。プログラム作成工程をいくつかの段階にわけて, 各段階ごとに進捗の検査と, 規程にてらしての質の検査を行なう。規程に反している場合は, 担当者に戻して訂正させる。

プログラマに納得のゆく仕様が示されていない場合は, プログラムを引受けるべきではない。オペレータやデータ処理を実行するグループは, コンピュータで処理するために必要な操作解説書や, プログラムの説明書が完備していなければ, プログラマからプログラムを引継ぐべきではない。

5.4 規準書の作成

新しいシステムを導入したり、開発しようとするときは、初めに規準書を作成して、それに従って作業を進める。そうすればはじめからそのシステムに関係する人達に規準を徹底することが出来る。最初からはっきりした規準を打出せないような領域があるときも、一応の決定をしておいて、後に不都合が生じた場合に変更するとよい。一応の決定も下せないときは、規準を決める日付を書込んでおく。

既存のシステムに、新しく規準を設定しようとするときは、次のような手順が考えられる。

- (1) 標準化委員会を結成する。できれば、経験豊かな人が2, 3人この仕事にあたることが望ましい。
- (2) 標準化の計画を発表する。会社の上層部が標準化に乘気で、バックアップしてくれることを確かめる。また、包含すべき領域、個々の規準、そこから得られる利点などについて、できるだけいろいろな意見を聞く。
- (3) 規準を作る項目の一覧表を作成する。これは、計画を作る第1歩としてどうしても必要である。
- (4) 項目一覧について、上司の承認を受ける。
- (5) 1項ずつ規準の草案を書きあげる。
- (6) 規準を更新する手順を決める。
- (7) 規準書を作りあげる。
- (8) スタッフを教育する。
- (9) 更新と保守を行なう。

既存のプログラムに手を入れて、最低の基準にかなうようにする。

5.5 規準書に盛込む内容

規準書には、次のような内容を盛込んでおく。

(1) 規準に関する方針

規準の適用範囲や、特に規準外の方法をとる場合の許可条件や許可の仕方、規準の管理、更新などについての責任部門などを説明する。

(2) 管理体制

管理機構と職務の明細、運営委員会、プログラミング部門の予算管理や原価計算、プランニングやプログラム作成の作業量の予測、教育訓練、報告の仕方、大きな組織のときは、スタッフの

電話番号なども盛込む。

プログラム作成の作業量の予測については、項を改めて具体的に触れたい。

(3) コンピュータの構成

コンピュータのハードウェアの構成と性能、ソフトウェアの機能と性能、文字セットなどについて詳しく記述しておく。

(4) システムの分析

サーベイ（現システムの調査）、問題の分析、システムの設計、開発、具体化などについての規準を盛込む。

サーベイや問題の分析については、いろいろ標準的なやり方が研究されているので、活用すべきである。チェック・リストを作っておき、チェック・リストと突合わせたり、監査手順を作ったり、監査したりすることが必要である。コンピュータで処理しようとする業務の内容の記述、プログラム仕様書やファイル説明書、処理手順についての記述、システムの流れ図などについて文書化しておくことが重要な仕事になる。文書化の技法を標準化しておくことによってシステム分析の結果は、体系的に掌握できるだろう。

(5) プログラミング

プログラミング作業中に承認を要する項目と承認の手続き、モジュールについての規準、プログラム中のラベルの付け方、リンケジについての約束、プログラムの管理、一般的なプログラミングの規則、プログラム・テストについての取決め、テストに必要な資料や用具、プログラムの修正、プログラムの要約、操作指示書その他マニュアル作成などについて盛込んでおく。

(6) 操作

コンピュータの使用について基準を取決めておく必要がある。一般的な規則、保守、カードの取扱い、コンピュータの操作、磁気テープやディスク・バック、ライブラリの管理などを盛込む。

5.6 プロジェクトの期間を推定する方法

参考になると思われるので、IBM発行の“データ処理規準作成の手引き”から抜粋して紹介する。

一般的考察

以下に述べるのは、EDPシステムが完成するまでに必要な延べ日数（プログラマとシステム分析員についての）を推定する方法である。この方法は、プロジェクトに要する時間を全体的に

掌握する上で、かなり正確な方法であり、プロジェクトの管理には欠くことのできないものである。また、この方法によれば、プロジェクトの作業ごとの必要時間がわかるので、仕事の予定表を作成し、仕事の割当てを行なうことが容易になる。

延べ日数の推定を行なうことによって、システム建設作業に内在する、いろいろな変量が明らかになる。この方法では、推定者は各変量に加重値を割当て、これらの加重値を延べ日数の算定に必要な要因とする。

この推定方法によって、システム建設における、次のような重要な作業にかかる延べ日数を知ることができる。

設計

開発

システムのテスト

設置とその後の処理

おのおのの作業の規定に関する仕事は、普通、プロジェクトの形成に先立って行なわれるものである。ここでは取扱わない。また、現在行なわれているサポートに関する仕事も適用できない。しかしながら、さらに研究を進めると、サポートの仕事自体をその仕事の目的に沿うように変更できる方法があるということも当然考えられる。

注意

推定の方法は、直感にもとづいたものであれ、公式化された一定の手順にもとづくものであれ、推定を行なうものの判断力によって良くも悪くもなる。ここで述べる技法にも、いくつかの欠陥があるので、推定者は、それに十分な注意を払わなくてはならない。まず、この方法は実際よりも正確であるような印象を与える。しかしながら、プログラマや分析員のイニシヤティブ、欲求、知能のような重要な変量が考慮されていない。また、この方法で考慮する要因は、固定された規則のように考えられて、実際の経験を反映しなくなる危険性もある。

一方、この方法には有利な点もいろいろあって、貴重な推定技術として使用できる。たとえば、この方法では、システムの開発上問題になる重要な変量を考慮しているし、推定過程の文書化、加えたい変更の評価なども行なっている。この方法によって割出された結果は、システム建設の日程表、およびプログラマ・分析員の動員の日程表を作成するうえで、重要な参考資料になる。さらに、手を加えて、実際の経験から得られたものを反映させるようにすれば、この方法は、必要な延べ日数を決定する非常に正確な方法になることは疑いない。

推定者としての前提条件

この推定法を使うためには、推定者は次のような知識を備えていなければならない。

- どんなシステムを開発するのか、その全体図。
- システムが処理するプログラムの入出力、ルーチンの種類、複雑度など。
- プロジェクトに配置されたプログラマやシステム分析員の経験の深さ。
- プロジェクト・スタッフに必要とされる業務に関する知識。
- 新しいプログラミング言語、新しいハードウェア、非常に複雑なレコードを単純化する仕事などに関する特殊な知識。
- 採用する加重システム。

加重値は、平均値をもとにして決める。システムによっては、非常に特殊な条件を考慮することもある。たとえば、頻繁に手操作で指示を与えなければならないような場合がこれである。そのような特殊な条件は、例外事項として推定に加えるようにする。

推定方法の8つのステップ

- (1) プログラムの複雑度を知る。
 - (a) プログラムの入出力の特徴を考慮して、それに重みづけをする。
 - (b) プログラムの主要処理機能に重みづけをする。
- (2) プログラミング技術を知る。
- (3) プログラムの業務に関する知識の程度を知る。
- (4) プログラムの開発に要する時間を知る。
- (5) システム建設にかかるその他の時間を知る。
- (6) 損失の因子を加える。
- (7) プロジェクト外の因子を加える。
- (8) 推定を完了する。

この方法は、いくつかの計算された推定を行なって、それを合計したものの方が、1つの全体的な推定よりも正確である、ということを前提としている。システムの設計、開発、テスト、設置に必要な延べ日数を知るには、まず、そのシステムの電子計算組織用のプログラムの開発に要する時間を割出し（ステップ1からステップ4まで）、それにその他すべての仕事を完了するのに必要な時間に加え（ステップ5）、さらに損失時間（ステップ6）と、プロジェクト以外に費した時間（ステップ7）を加える。これによって得られた結果が、システムの建設に配置されたプログラマとシステム分析員が働く延べ日数である（ステップ8）。

別 紙

ソフトウェアに関する書類規定およびその解説

1 目 的

書類は、開発するソフトウェア・システムに関して、その内容ならびにKnow-howを記述したものであり、書類によって、成果を確認するものである。

2 範 囲

公式書類とは、6～10項に述べる下記の書類をいう。

- (1) 構成および機能仕様書
- (2) 設計仕様書
- (3) ソース・リスト
- (4) 検査成績書
- (5) 利用者マニュアル

3 工程管理の体制

書類作成に関する管理は、ソフトウェアの工程管理と密接な関係があり、書類管理の確実な実行を期するためには、工程管理についての方法、組織などの体制の確立を前提としなければならない(逐条解説参照)。

4 作 成

書類は以下に示す時期に、作成しなければならない。

作 業 段 階	計 画	設 計	製 作	検 査	保 守
書 類					
利用者マニュアル			○		
機能および機能仕様書	○	○			
設計仕様書		○			
ソース・リスト				○	
検査成績書				○	

5. 変 更

書類に変更の必要が生じた場合には、書類を更新せねばならない。

変更に関する手続きは、作製の場合に準ずる。

6. 構成および機能仕様書

システムならびにその構成単位であるサブシステムなどについて、設計思想、機能、構成などの記述を含む構成および機能仕様書（以下CF仕様書と略称する）を作成する。

6.1 作成の目的

CF仕様書作成の目的には、大別して次の2つの面がある。

- (1) System Subsystem Module, Procedureなど、上位の機能単位から、下位の機能単位へ、階層別に当該機能単位の機能を決定してゆき、それらの機能仕様書の無矛盾性、有効性、実現可能性を検討し、最終的にはそれらの保証のもとに設計仕様書を作成するものである。
- (2) システム全体の機能と構成の関係を、前述のような階層別に、その概観（overview）を記述し、これらの外観が、少なくとも木構造に配列されていて、システム全体を容易にかつ組織的に概観できるようにするものである。

6.2 記述の方針

6.1(1)は、広義の設計作業そのものであり、従来この過程が確実に実行されず、かつ組織化に欠けていたために、設計仕様書作成の段階で時間を浪費する傾向があったと思われる。

6.1(2)の必要性は、実際の設計作業においては、グループ内または関連するグループ間での情報交換を緊密にし、また設計者の異動などの場合にも、迅速に処置がとれる点にある。

さらに、設計作業に続く、検査ならびに保守においては、このような仕様書の存在は設計者以外の者がシステムを理解するための不可欠な条件となる。

このような、2つの要求を完全に一致させるようなCF仕様書の記述は容易ではないが、本規定では、まず前者の目的にもとついて記述し、設計の進むにつれて、後者の要求にも沿うように、記述を補っていくことを要望する。

6.3 作成と納入の関係

前述したような階層ごとの仕様の記述は、CF仕様書全体の構成からみて、大項目、中項目、

小項目（または章，節，項など）の形式で，逐次作成され納入されなければならない。

6.4 記述の内容

それぞれのC F仕様書は，概要以下のような内容を含むものである。

- (1) 6.1 項に述べたような趣旨で，対象となるプログラムが全体として果す機能，外部仕様（入出力データ），および論理的な構成
- (2) 上記の仕様を決定するに至った設計方針
- (3) 6.1 項(2)に述べるような意味で，上位のC F仕様書の記述との関連
- (4) デバグに関する方針

6.5 記述の様式

C F仕様書は，文章による設計概念の記述とともに，ブロック図，総括的なフローチャートなどの説明図を含むものとし，A 4 判標準用紙に記述する。A 4 判1 ページの標準字数は30×25 字程度とする。

6.6 C F 仕様書の見出し部

C F仕様書には，下記の項目を含む見出し部を付ける。

- (1) 公式書類番号（下2けたは改訂番号とする）
- (2) 当該機能単位の名称
- (3) 作成年月日
- (4) 作成者氏名，所属
- (5) 改訂履歴（書類番号，年月日，氏名，所属，改訂内容を列記する）

6.7 関連記述の冗長さの必要性

当該システムと関連し，当該システムの設計概念を理解するに必要な他システムの内容は，簡易明確に記述した説明書として，C F仕様書に添付する。

7 設計仕様書

設計仕様書は，適当な機能単位ごとに，その機能単位の詳細な設計作業に必要な下記(1)，なら

びにその製作作業に必要な下記(2)の内容を含むものとする。

(1) 上位のC F仕様書と下記の設計仕様に関連を示し、最下位のレベルのC F仕様書に相当する記述

(2) 7.1～7.5に詳述するprocedureごとの設計仕様

各procedureの設計仕様書は、文章ならびにフローチャートの両方で記述し、その内容は下記のsectionを含むものとする。

フローチャートを除く設計記述の文章は、8.4のコメントを除くソース・リスト1ページ当たり最低A4判1ページ以上とする。

7.1 Identification Section

- (1) 公式書類番号(下2けたは改訂番号とする)
- (2) procedureの名称
- (3) 当該procedureが属するサブシステム、モジュールなど上位プログラム単位の名称
- (4) 記述言語
- (5) 作成年月日
- (6) 作成者氏名、所属
- (7) 検査ならびに保守責任者氏名、所属
- (8) 改訂履歴(書類番号、年月日、氏名、所属、改訂内容を列記する)

7.2 Interface Section

- (1) すべてのentry point およびreturn pointのリスト
- (2) 各entry point およびreturn pointにおけるすべてのパラメータのtypeならびにその値の割当て
- (3) linkage に関する特記事項
- (4) 当該procedureがcallする他のprocedure全部の名称とその公式書類番号
- (5) 当該procedureをcallする他のprocedure全部の名称とその公式書類番号
- (6) 当該procedureに関係するテーブル類とそのテーブル類の更新の手続き

7.3 Logic Section

- (1) 3～30行程度のプログラムの目的、機能の概要

- (2) procedure内の手法ならびに特徴を簡易明確に記述する説明(説明図を含む)
- (3) 各entry point およびreturn pointにおける入出力パラメータ全部の意味の説明
- (4) コーディングに関する特記事項
命令語のアドレス部以外を変更するなど、きわめて特殊なコーディング技術を使用する場合には、必ず詳細な説明とその説明図
- (5) エラー処理に関する特記事項
特定の条件で計算機が一時停止するようなエラー処理を行なう場合には、その条件の持つ論理的な意味と、それに関する計算機各部の表示灯の状態

7.4 Debug Section

- (1) デバグに使用するすべてのダミ・モジュールの目的と機能の概要
- (2) システム・シミュレータ, その他のプログラム開発手段との関連
- (3) 設計者が必要と認めるチェック項目およびその内容説明
- (4) チェック困難な項目のリスト
- (5) パラメータの代表的な組合せについての実行時間の推定値と実行条件
- (6) 主記憶占有容量

7.5 Flowchart Section

フローチャートまたはこれに類する記述用チャートは, 7.3 Logic Sectionの記述と相補的なものとし, 当該procedureの論理を誤りなく読取れるように記述する。

コーディングの2〜3ステップをフローチャートの1個のボックスに対応させるように細部まで詳細に記述したいいわゆるdetailed flowchartは含まない。

記述に際しては, 下記の規約に従う。

- (1) フローチャートで使用する略号で, Logic Sectionで説明していないものは全部余白に注記すること。
- (2) フローチャート・シンボル(◇や○など)の意味を標準化し, かつパターン定規を用いるか, または機械的手段を採用して, すべてのフローチャートを書くこと。
- (3) フローチャートで使用する記号・略号は原則として, Logic Sectionで説明したものと同一であること。

- (4) 1つのフローは、2ページ以内におさめること。あるフローの一部をさらに詳細に示すフローを書く場合には、相互の関係は両者の余白に注記し容易に対照できるよう考慮すること。
- (5) フローチャートのterminal label およびblock label はソース・リストのstatement label と一致させるよう考慮すること。
- (6) フローチャートはA3またはA4判標準用紙に記述すること。

8 ソース・リスト

各procedureは、原則としてアセンブラ言語によるコーディングの場合には100ないし200ステップ程度に、作業用言語による場合にはコメントを含め、ソース・リストで1～20ページ程度に分制作製する。

ソース・リスト1ページの標準行数は66行（適宜の空行を含む）とする。

8.1 作業用言語のコメント

作業用言語によるコーディングの場合には、平均してstatement 1行当たり1行程度のコメントをそう入すること。

外部変数の宣言には必ずコメントを付け、内部変数の宣言にも、必要に応じコメントを付けること。

機械語をビット処理するstatement、I/O関係などで、コントロール語を作成または変更するstatementには、特に平均して1行当たり2行程度のコメントをそう入する。

8.2 アセンブラ言語のコメントおよびマーク

アセンブラ言語によるコーディングの場合には、各命令の余白に必ずコメントを書き、なお適当なブロックごとにそのブロックに関する説明をリマークとしてそう入すること。

なお、7.3(4)に記述した特殊なコーディングを行なったときには、設計仕様書中の説明の要約をリマークとしてそう入すること。

8.3 コメントおよびリマークの表現

コメントおよびリマークは、だれにでもわかりやすいようにできるだけ統一的な表現（たとえば作業用語に類似した表現）で記述すること。

また設計仕様と必ず対応するよう考慮すること。

8.4 ソース・リストの総括コメント

検査を終了したソース・リストの初めに、下記の項目内容をコメントの形でそう入すること。

- (1) 最新の設計仕様書の identification section
- (2) 最新の設計仕様書の interface section
- (3) 検査済みであることの証明

9 検査成績書

検査責任者（担当者）は検査を行なう前に、少なくとも設計仕様が正確に実現され利用者マニュアルの記述が確実に実行可能であることを保証する機能検査のための検査手順、検査条件などを含む検査仕様を記述した検査仕様書を作成する。

検査の完了後に、検査の結果を簡明に判定できる成績書を作成し、実際に行なった検査結果から、完全に改訂した前記検査仕様書を添えて、検査成績書とする。

9.1 検査仕様書

完全に改訂した検査仕様書は下記の項目の記述を含むものとする。

- (1) 公式書類番号（下2けたは改訂番号とする）
- (2) 検査したプログラム単位の名称
- (3) 検査終了年月日
- (4) 検査責任者（担当者）氏名
- (5) 検査のためのテスト・プログラムの内容ならびにデータの概要
- (6) 検査条件
 - (a) 検査の対象としたすべてのプログラムのリスト
 - (b) 検査に使用したすべてのダミー・モジュールのリスト
 - (c) 検査に使用したハードウェア構成およびそれらの set up
- (7) 検査の対象となるすべてのプログラムについて、検査の参考となる検査結果の概要、ならびに検査に使用したすべてのダミー・モジュールの機能の概要（重複を避けるために、過去に検査を完了した検査仕様書を参照することができる）

9.2 成績書

成績書の内容は、検査結果を簡明に判定できるような下記項目を含むものであって、必要な場

合を除き、いたずらにダンプ結果をもって代替しないこと。

- (1) 入力データのリスト
- (2) 出力データのリスト
- (3) 動作状態の記録
- (4) 検査時の `console log sheet`

10 利用者マニュアル

内容の記述形式については特に規定しない。従来の慣行による。ただし、必ず索引を付けること。なお、CF仕様書の見出し部に準拠した見出し部を付けること。

Ⅲ 情報処理用流れ図記号

JIS C 6270—1970

昭和45年4月1日制定

情報処理用流れ図記号

1 適用範囲

この規格は、情報処理のための流れ図記号 (flowchart symbol) とその用法について規定する。

備考 流れ図記号は、情報処理系における操作の系列、データの流れ、作業の進行などを表現するためのものである。したがって、絵や図を用いて情報処理系を描写する絵画的なものについては規定しない。

2. 用法の一般事項

2.1 流れ図 (flowchart) における流れ (flow) の方向は、原則として、左から右へ、上から下へとする。流れの方向がこれに合わないときには、流れを示す矢印を用いなければならない。なお、矢印を用いた方がわかりやすいときには、これを用いることがのぞましい。

2.2 流れ線 (flow line) は互いに交差してもよい。この場合には、これらの間に互いに論理的関係はないものとする。

2.3 2つ以上の流れ線を集めて、1つの流れ線にして出してもよい。

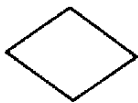
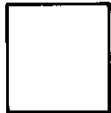
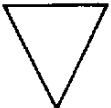
2.4 記号は、直ちには識別できないほどにまで、形を変えたり回転したりしてはならない。

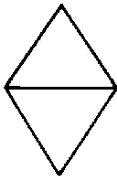
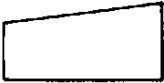
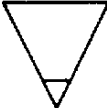
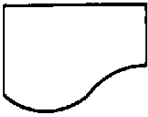
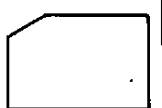
備考 記号の大きさや縦横の比率についてはとくに規定しない。記号は、実線によることを原則とするが、とくに必要な場合には破線などを使ってもよい。

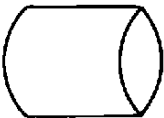
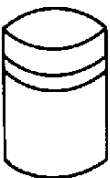
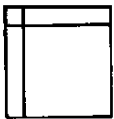
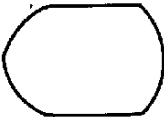
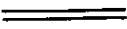
3 流れ図記号

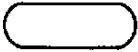
流れ図記号は、つぎのように定める。

番号	記号	意味
1		処理 (process) あらゆる種類の処理機能を表わす。

番号	記号	意味
2		判断 (decision) いくつかの択一的経路のうち、どの経路をとらせるかを決める判断、またはスイッチ式の操作を表わす。
3		準備 (preparation) スイッチの設定、指標レジスタの変更、ルーチンの初期値の設定など、プログラム自身を変えるための命令、または命令群を修飾する機能を表わす。
4		定義済み処理 (predefined process) サブルーチンなど、別の場所で定義されている命令群、またはいくつかの操作からなる命名された処理過程を表わす。
5		手作業 (manual operation) 機械的な手段を用いない手作業によるオフラインの処理過程を表わす。
6		補助操作 (auxiliary operation) 中央処理装置で直接に制御されていない装置によって行なうオフライン操作を表わす。
7		組合せ (merge) 記録の2つ以上の集合を1つの集合に組合わせることを表わす。
8		抽出 (extract) いくつかの記録からなる1つの集合のなかから、特定のいくつかの集合を取分けることを表わす。
9		照合 (collate) 組合わせたり抽出したりしながら、記録の2つ以上の集合から2つ以上の集合を作り出すことを表わす。

番号	記号	意味
10		分類(sort) 集合中の記録をある定められた順序に並べ替えることを表わす。
11		手操作入力(manual input) オンラインけん盤, スイッチ, 押しボタンなどを手で操作して, 処理中に情報を入れる入力機能を表わす。
12		入出力(input/output) 情報を処理可能にする入力機能, または処理済みの情報を記録する出力機能を表わす。
13		オンライン記憶(online storage) 磁気テープ, 磁気ドラム, 磁気ディスクなど, あらゆる種類のオンライン記憶を利用する入出力機能を表わす。
14		オフライン記憶(offline storage) 情報を任意のオフラインの記録媒体に記憶する機能を表わす。
15		書類(document) 書類を媒体とする入出力機能を表わす。
16		紙カード(punched card) 紙カード(マークセンス・カード, パーシャル・カード, スタブ・カード, マーク・スキャン・カードなどを含む)を媒体とする入出力機能を表わす。
17		カードの組(deck of cards) 紙カードの集りを表わす。
18		カードのファイル(file of cards) 紙カード上の関連ある記録の集りを表わす。

番号	記号	意味
19		紙テープ (punched tape) 紙テープを媒体とする入出力機能を表わす。
20		磁気テープ (magnetic tape) 磁気テープを媒体とする入出力機能を表わす。
21		磁気ドラム (magnetic drum) 磁気ドラムを媒体とする入出力機能を表わす。
22		磁気ディスク (magnetic disk) 磁気ディスクを媒体とする入出力機能を表わす。
23		磁心 (core) 磁心を媒体とする入出力機能を表わす。
24		表示 (display) 処理中に、オンライン表示灯、映像表示装置、操作卓の印字機、作図機などを通じて、情報を人間が利用できるように表示する入出力機能を表わす。
25		流れ線 (flow line) 記号を結びつける機能を表わす。用法は2.1による。流れ線の交差は2.2による。流れ線の合流は2.3による。
26		並行処理 (parallel mode) 2つ以上の同時操作の始り、または終りを表わす (左図には流れ線は示していない)。
27		通信 (communication link) 情報を通信線で伝達する機能を表わす。

番号	記号	意味
28		結合子 (connector) 流れ図のほかの場所への出口, あるいはほかの場所からの入口を表わす。
29		端子 (terminal, interrupt) 開始, 終了, 停止, 遅延, 中断など, 流れ図の端子を表わす。
30		注釈 (comment, annotation) 明りょうにするために, 説明または注意を加える機能を表わす。

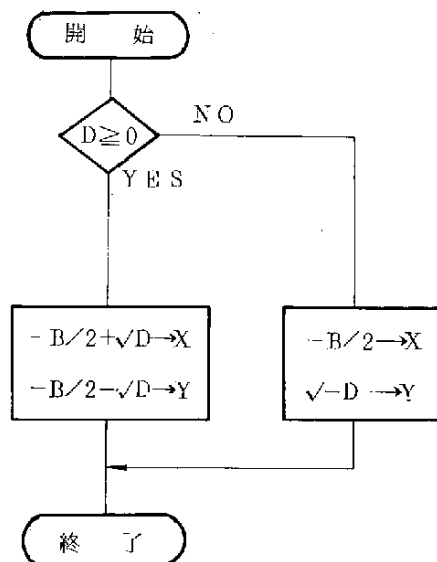
4 流れ図中の記述

4.1 本文

流れ図記号に付ける本文 (text) は, 流れの方向にかかわらず, 左から右へ, 上から下へ読むように書く。

備考 本文は, 可能な限り記号の内部に記入することがのぞましい。

例: 図1



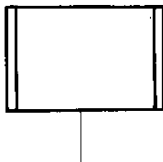
4.2 識別名

流れ図記号に付ける識別名 (symbol name) は, プログラムなどを参照できるように記号

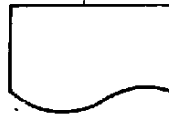
に名前を付けるもので、その最大字数をあらかじめ定め、その範囲内で記号の左上に書く。

例：図 2

FIXEDNUM



NOTOVFL

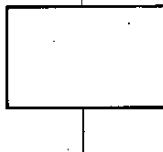


4.3 説明

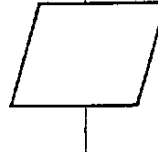
流れ図記号に付ける説明 (symbol description) は、流れ図が表わす系のなかでの、その記号の機能をよりよく理解できるようにするために記載する、前後参照など、あらゆる情報である。一般に任意の字数の文字を用いて、記号の右上に書く。

例：図 3

NOTOVFLのための準備



F 7



5 結合子

5.1 出結合子

流れ線が中断される点を表示するのに用いる結合子を、^で出結合子 (outconnector) と呼ぶ。

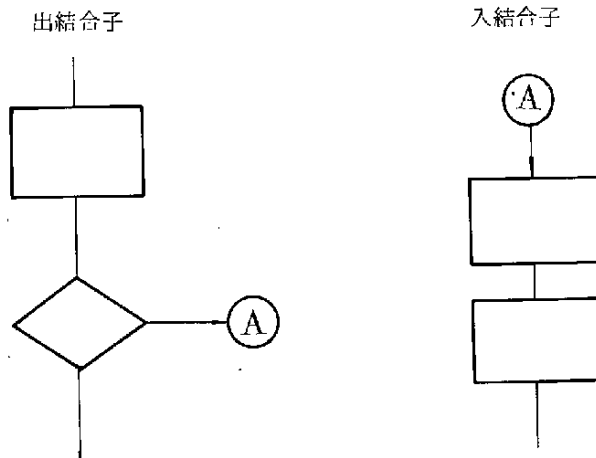
5.2 入結合子

中断された流れ線が再開される点を表示するのに用いる結合子を、^{いり}入結合子 (inconnector) と呼ぶ。

5.3 結合

出結合子およびそれに対応する入結合子のなかに、同じ文字、数字、名前などの識別子 (identifier) を記入し、これらが結合されていることを表わすものとする。

例：図 4



6. 横線

6.1 機能

流れ図記号のなかに書きこまれた横線は、この記号の機能に関する、より詳細な表現が同じ流れ図中のほかの場所に記載してあることを示す。

備考 定義済み処理記号においては、横線の書きこまれた記号とは異り、その機能に関する、より詳細な表現が同じ流れ図中にある必要はない。

6.2 書き方

横線は、流れ図記号の上縁に近い内側に水平に記入する。また、この記号の機能を詳細に表現した場所に対する識別子を、記号の上縁と横線の間に記入する。

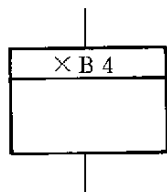
備考 横線を記号の左右の縁につくまで十分に延ばすかどうかは、規定しない。

6.3 詳細表現の参照

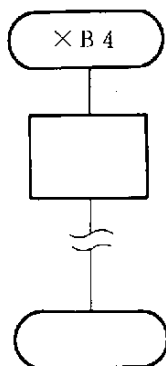
横線の書込まれた流れ図記号 (striped symbol) の機能を詳細に表現した場所では、その始りと終りとに端子を用いる。始りの端子のなかには、6.2に従って記入した識別子と同じ識別子を記入する。

例：図 5

横線の書込まれた記号



詳細な表現



7. 2つ以上の出口

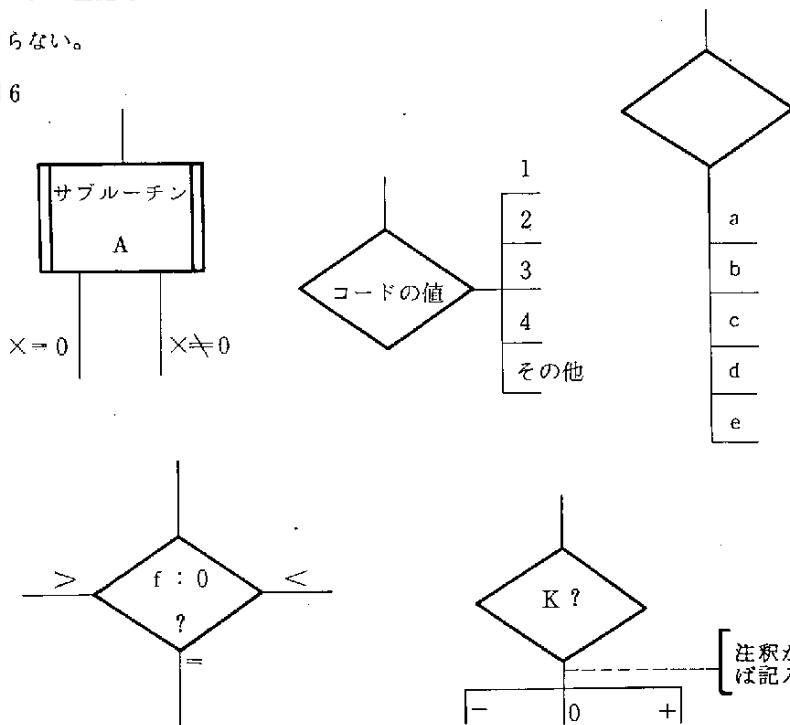
7.1 分岐の表現

1つの流れ図記号からの出口を2つ以上書く場合には、必要な数だけの流れ線をその記号から出すか、あるいはその記号から出た流れ線が必要な数だけに分岐する形で表現する。

7.2 分岐条件の記入

1つの流れ図記号に2つ以上の出口がある場合には、それぞれの出口に分岐条件を記入しなければならない。

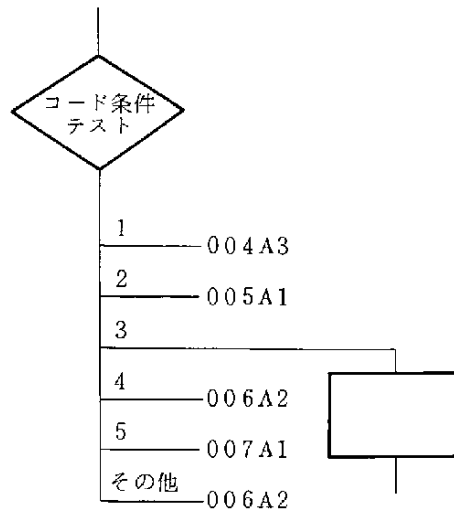
例：図 6



7.3 分岐表

7.2で規定した分岐条件は、それに関連した入結合子への識別子と組にした形などで表現してもよい。

例：図7



(注) 1, 2, 3, 4, 5, その他は, 条件を表わし, 004A3, 005A1などは入結合子への識別子を表わす。

8 同種類の媒体の反復表現

8.1 反復

多数の写し, 各種の印刷形式の報告書, 各種のせん孔形式の紙カードなど2つ以上の媒体やファイルを使用したり作成したりすることを示すには, 必要な本文を記入した入出力用の単一の流れ図記号を用いるかわりに, 同じ記号を重ねて書表わしてもよい。

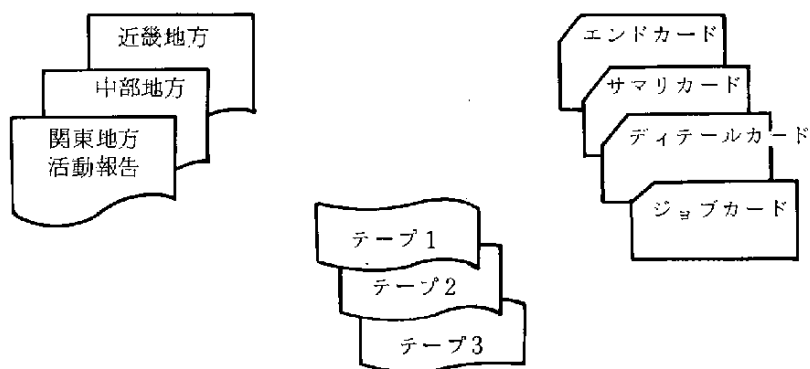
8.2 位置

重ねて書く場合には, 先頭の流れ図記号は完全な記号の形式で書く。そのうしろへ同じ記号を反復するには, 先頭の記号から上または下, かつ右または左にずらして2番目の記号を書く。3番目以降の記号についても, 同じ方向にずらしながら書かなければならない。

8.3 順序

重ねて書いた流れ図記号が順序の意味を含んでいる場合には, その順序は, 前(先頭)からうしろ(末尾)に向かうものとする。

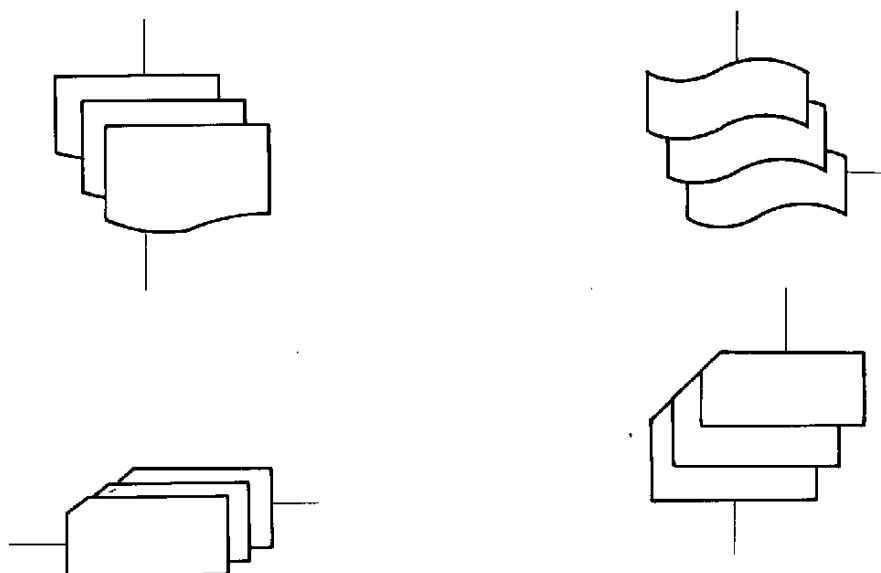
例：図 8



8.4 流れ線

流れ線は、重ねて書いた流れ図記号のどの点にはいり、また、どの点から出てもよい。重ねて書いた記号について 8.3 で定めた順序は、流れ線の出入りの点のいかんによっても変更されない。

例：図 9



情報処理用流れ図記号 解説（抜粋）

制定の要点

情報処理用流れ図は、情報処理系での操作の系列、データの流れ、作業の進行などを図式化したものである。流れ図は、図式化することによって問題の解析、人対人の伝達をよくすることを目的としており、情報処理の分野で広く使用されているが、従来作成基準が明確でないため、不統一な流れ図が作られたり、誤解を生むことがあった。これを防ぐためには、標準化を行なう必要があり、国際的な標準化の動きを考慮しながら、昭和43年度、(社)情報処理学会に日本工業標準原案調査作成を委託し、この原案をもとに、昭和44年度に日本工業標準調査会の審議を経て制定をみるに至った。

国際的な標準化の経緯

国際標準化機構(International Organization for Standardization : ISOと略称する)のTC97/SC7で流れ図記号とその用法の標準化が行なわれており、すでに流れ図記号については、昭和43年6月、アムステルダムで開かれたTC97第5回総会で「流れ図記号の文書(97N90)」が承認され、その後、ISO推薦規格(ISO/R1028-1969)となった。一方、流れ図記号の用法については、昭和43年6月のアムステルダムにおけるTC97/SC7会議の決定に従って第1次規格原案「用法に関する文書(97/7N56)」が作成され、各国から賛否に関する投票が行なわれた。その結果、賛成多数であったが、なお記述上の意見がいくつかあり、最終的調整が残されている。わが国の国内委員会も上記国際的な動きに対し、委員の出席、資料の提供により随時意見を表明し貢献している。

制定の基本方針

この規格は、つぎの方針でまとめている。すなわち、

- (i) 国際標準と矛盾せず、しかもなるべくそれと対応がつくよう文字、表現、全体の構成をまとめる。
- (ii) 機器の標準とはその利用方法が異なるので、使用上の不都合、誤解のおこらぬ範囲で利用上の

自由度をもたせるよう配慮する。

上記(i)の方針により、ISOの2つの規格の構成はほぼ対応しており、「流れ図記号の文書(97N90)」は第1～3章に、「用法に関する文書(97/7N56)」は第4章にそれぞれ相当している。さらに、個々の対応関係を明らかにするため、この規格中にある主要な術語は、ISO文書での対応英語が、かっこ内に記入してある。

1 適用範囲

備考に「したがって、絵や図を用いて情報処理系を描写する絵画的なものについては規定しない」とあるのは、この規格で定める記号は、電子計算機を使用して情報処理を行なう場合に必要なシステム・デザインやプログラミングなどに用いる流れ図記号であって、たとえば模式的な図形などを規定したものではない、という意味である。

2 用法の一般事項について

2.4およびその備考は、流れ図記号の大きさとか、縦横の比率については使用者にまかせられているが、あまり自由に変えないほうがよいという意味である。少なくともこの規格の記号欄の流れ図記号は、ほぼ、標準的な形、図形相互の相対的な大きさを示したものであるもので、これを参考にすることがのぞましい。なお、この規格票において記号を作図するのに用いた手法を参考としてのちに示す。

3 流れ図記号について

番号1の処理(process)は、たとえば情報の値、形、場所などを変更したり、またはいくつかある流れの方向のどれかに決めたりする操作、もしくは操作群の実行を表わす。

番号4の定義済み処理(predefined process)は、プログラム作成の際、同一手順の処理過程をある条件のもとに何回も実行したい場合に用いる。このような場合には、サブルーチンを作り、それには、必要ときいつでも照合できるように名前が付けられる。このようなサブルーチンの記述、あるいはマクロのようにあらかじめ定められた既製の処理過程を表わす場合などに用いる(解図8、解図9参照)。

番号5の手作業(manual operation)は、「機械的な手段を用いない手作業」とあるが、たとえば手によるカードの抜き出しなどの作業がこれに相当する(解図10、解図11参照)。

番号16の紙カード(punched card)は、英語の記号名punched cardに対する日本語としてあえてせん孔カードを採用せず、紙カードとしたが、それは、せん孔カードではないマークセンス・カードなどを含めたためである。

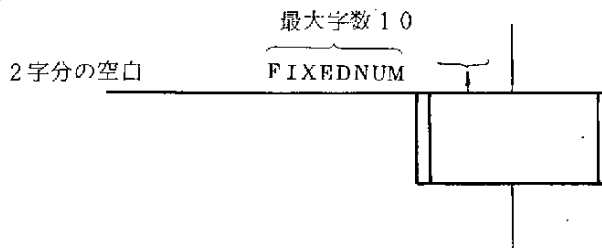
番号17のカードの組(deck of cards)と番号18のカードのファイル(file of cards)との相互関連については、多少不分明なところもある。ファイルを強調する場合には18を使用するが、17を用いてもかまわない。17の場合には、どちらかといえば、ファイルよりも紙カードの集りという物理的性質が強調される意味合いをもつ。このへんの使い方は、この規格作成にあたって参考にしたISO推薦規格の同種の記号においても弾力的な運用を認めている。

4 流れ図中の記述について

4.1本文(text)は、各種の流れ図記号に、その処理、操作などの内容を付記するときの規定である。本文の書き方は、流れの方向に読むのではなく、つねに左横書きに従うことを規定している。

4.2識別名(symbol name)は、流れ図記号に名前をつけておき、プログラム・シートなどを参照するための便を考えた規定である。この目的のための名前を識別名と呼び、その流れ図への記入の方法は、流れ図全体を通じて識別名の最大字数を定めておき、流れ図記号の左上に、その字数の範囲内で記入することとしている(最大字数に満たない識別名は、「空白」を含めて、つねに最大字数になっていると考えてよい)。

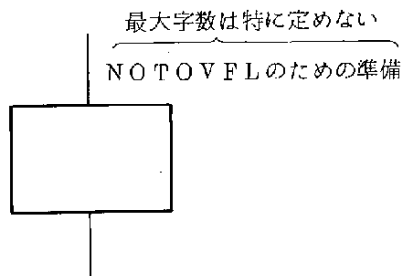
解図1



4.3説明(symbol description)は、本文の追加的な情報や、流れ図中の前後参照などの目的で流れ図記号に付ける情報(本文および識別名以外)の記入に関する規定である。この情報を説明と呼び、その流れ図への記入方法は、特に最大字数を定めず、流れ図の右上に記入

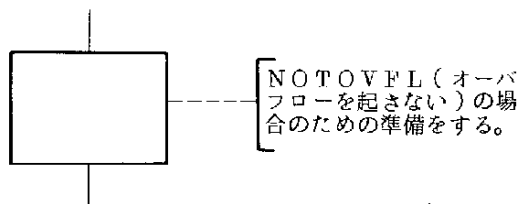
することとしている。

解図 2



なお、記入すべき説明が長文であったり、注釈的なものであったりする場合には、この記述法によらず、注釈記号を用いるほうがよい（解図 8，10，11 参照）。

解図 3



5 結合子について

これは、結合子記号の用法の規定である。5.1で出結合子（outconnector）を、5.2で入結合子（inconnector）を定義し、5.3で具体的な用法を規定している。流れ図を書くときには、ある流れ図記号から出た流れ線が、流れ図の同一ページ中のほかの離れた場所につながるような場合や、あるページの流れ図中の部分からほかのページの流れ図中の部分に連結するような場合がある。このような場合に、出る部分とはいる部分の流れ線の先端に結合子を付け、この記号のなかに同じ識別子（identifier）を記入して、相互に連結していることを示す（解図 11，12 参照）。

6 横線について

6.1 機能は、流れ図中で、詳細な記述を別に分けて書きたい場合の用法を規定したものである。

たとえば、開いたサブルーチンを含むプログラムを、主ルーチンとサブルーチンとに分けて記述するときなどに、この横線の用法が応用できる。

簡単な例として、解図 9 にあげたものは、ハウスキーピング処理を、横線の書込まれた処理記号で表現しておき、同じ流れ図中のほかの場所に、このハウスキーピング処理だけの独立した詳細な流れ図を書いたものである。このとき、1つのハウスキーピング処理の識別子を“INITIATE”として横線の上に書き、この詳細表現を行なう独立した流れ図は“INITIATE”という同じ識別子を記入した端子記号から始っている。また“TERMINATE”も同様に用いられている。

6.1の備考は、横線の書込まれた処理記号と定義済み処理記号との関係を述べたものである。

解図 4



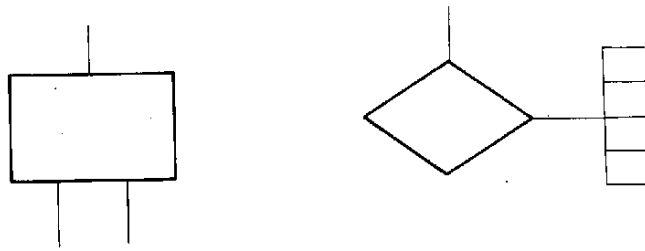
両者は、機能的に非常に似ているが、定義済み処理記号は、すでに一般的に定義されているか、またはほかの流れ図でまえもって定義されている独立した部分的な処理（閉じたサブルーチンなど）を表現するときに適用されるものである。一方、横線の書込まれた処理記号は、概要を書いた流れ図との関係などを示すことを、おもに想定して設定したもので、前者とやや異った意味合いを持つものである。

7. 2つ以上の出口について

流れの中間にある流れ図記号には、その流れ図記号にはいる流れ線と、それから出る流れ線が存在する。後者の流れ線が2つ以上ある場合には、なんらかの条件によって流れが分岐することを表わしている。

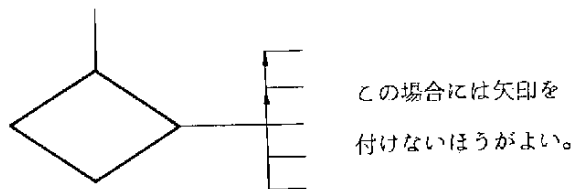
7.1では、その分岐する流れ線の表現法を規定し、7.2では、その分岐条件の表現法を規定している。分岐する流れ線の表現法は、流れ図記号から直接2つ以上の流れ線を出す方法と、流れ図記号から出た1本の流れ線を2つ以上に分ける（分岐する）方法とがある（もちろん、この両者の組合せもある）。

解図 5



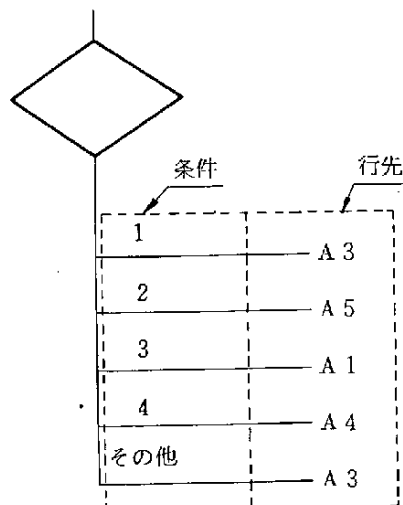
とくに後者の場合には、分岐の流れ線に、左から右、上から下への流れの方向の原則に合わないものが部分的にできるが、流れ図を不必要に複雑にしないために、矢印を省略することがのぞましい。

解図 6



分岐条件の表現は、7.2の例にあるように、流れ図記号に近く、2つ以上の分岐した流れ線の右（縦方向の流れ線のと看）、または上（横方向の流れ線のと看）にそれぞれ記入することが多い（解図10、11参照）。また7.3には、解図7のように、分岐した流れ線を流れ図記号（端子記号を含む）に接続せず、分岐条件と行先の入結合子への識別子を表の形のように並べる用法を許すことが述べられている（解図11参照）。

解図 7

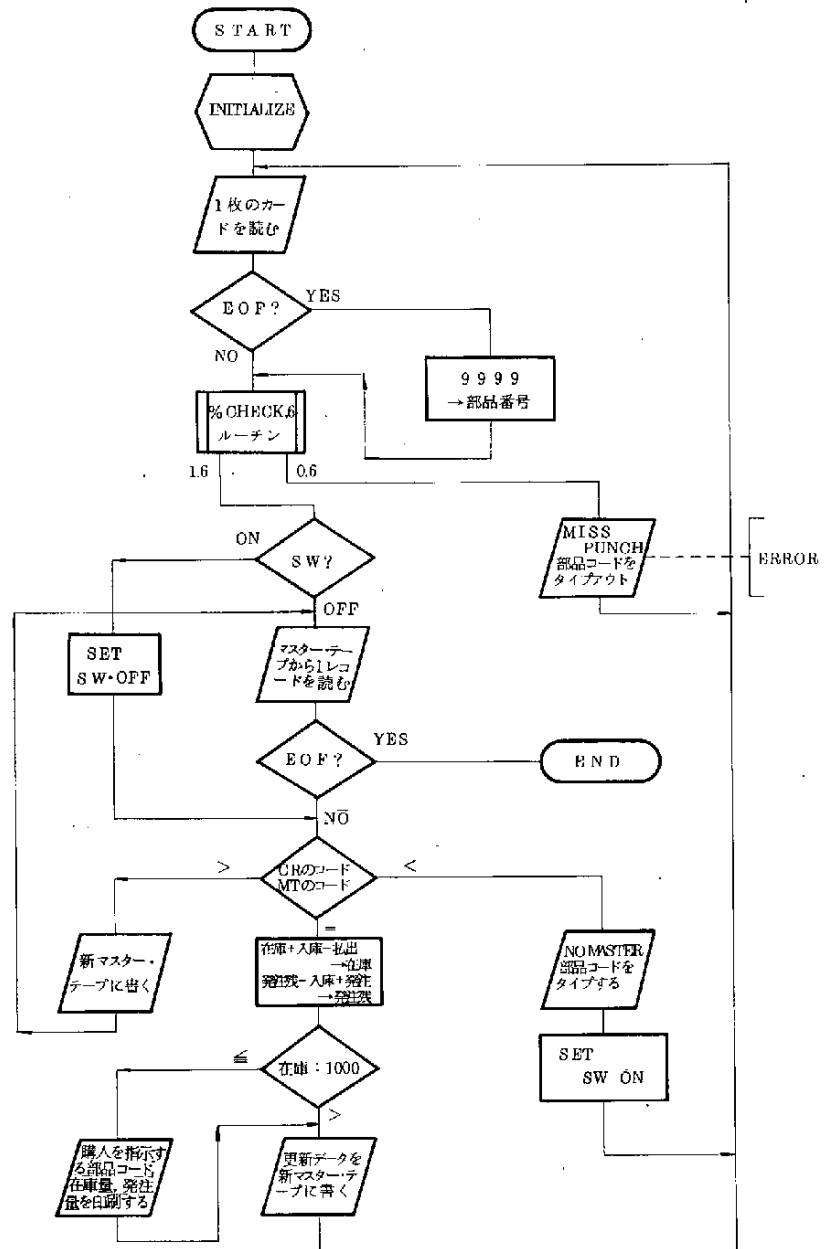


8 同種類の媒体の反復表現について

各種のファイルを流れ図中に表現する場合、同種類の媒体の2つ以上のファイルを同時に記述する必要が考えられる。この場合の流れ図記号の重ね合せ表現についての規定である。8.2で重ね合せ方の位置、8.3で重ね合せ方の順序、8.4でこの場合の流れ線との接続法について、それぞれ規定している（解図10参照）。

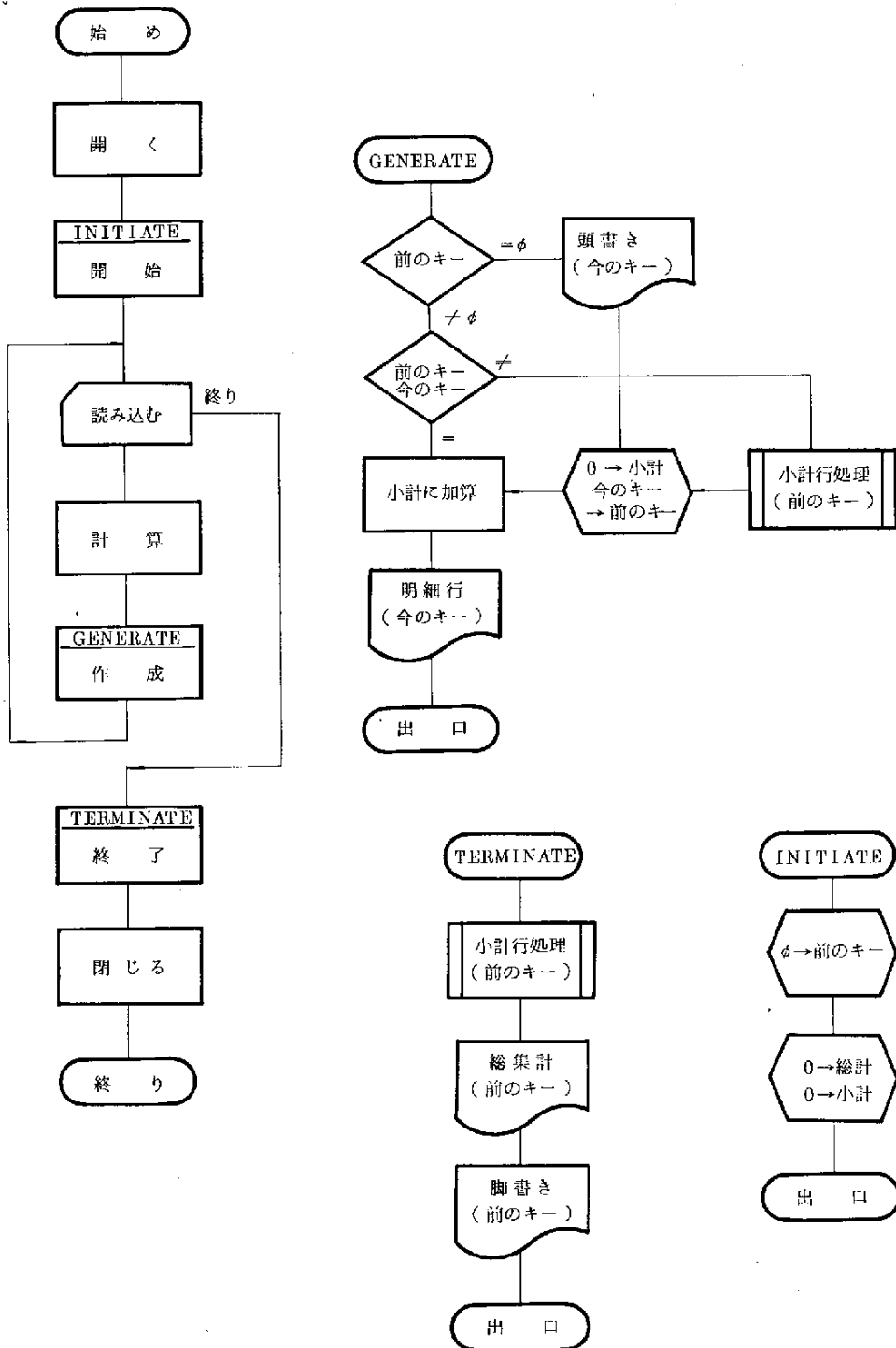
解図 8

在庫管理用流れ図の例



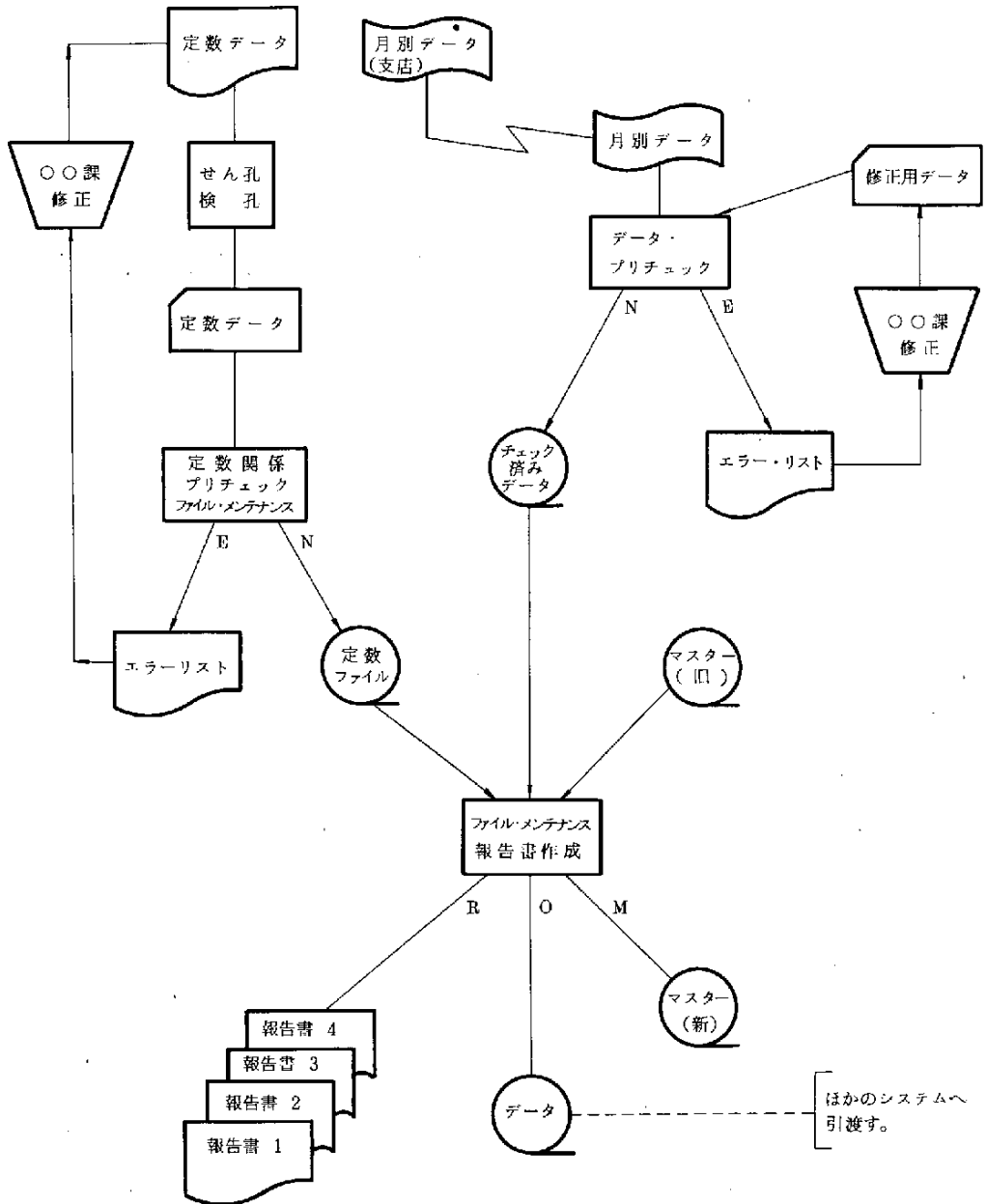
解図 9

報告書作成プログラム流れ図の例



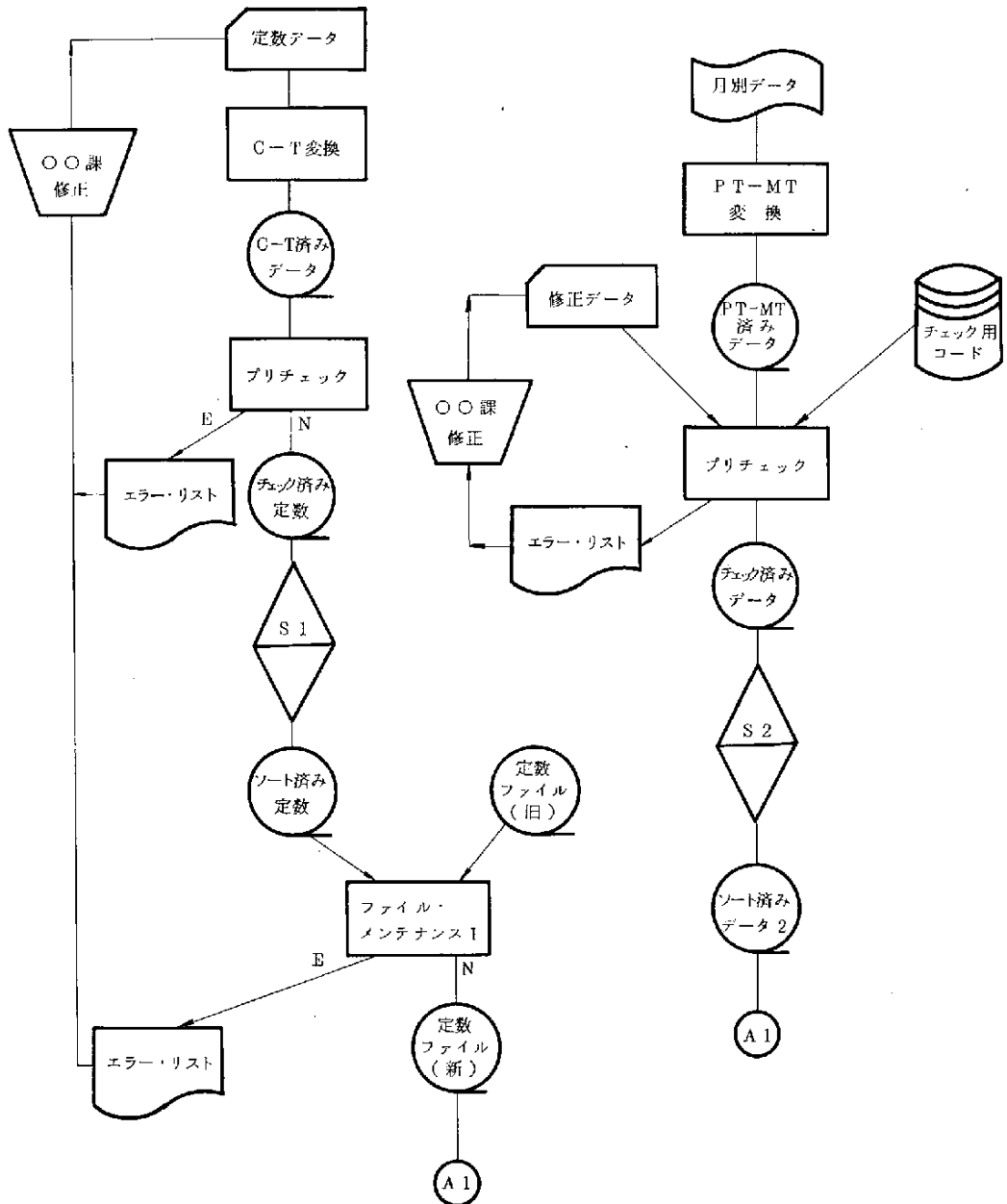
解図 10

月報作成システム流れ図の例

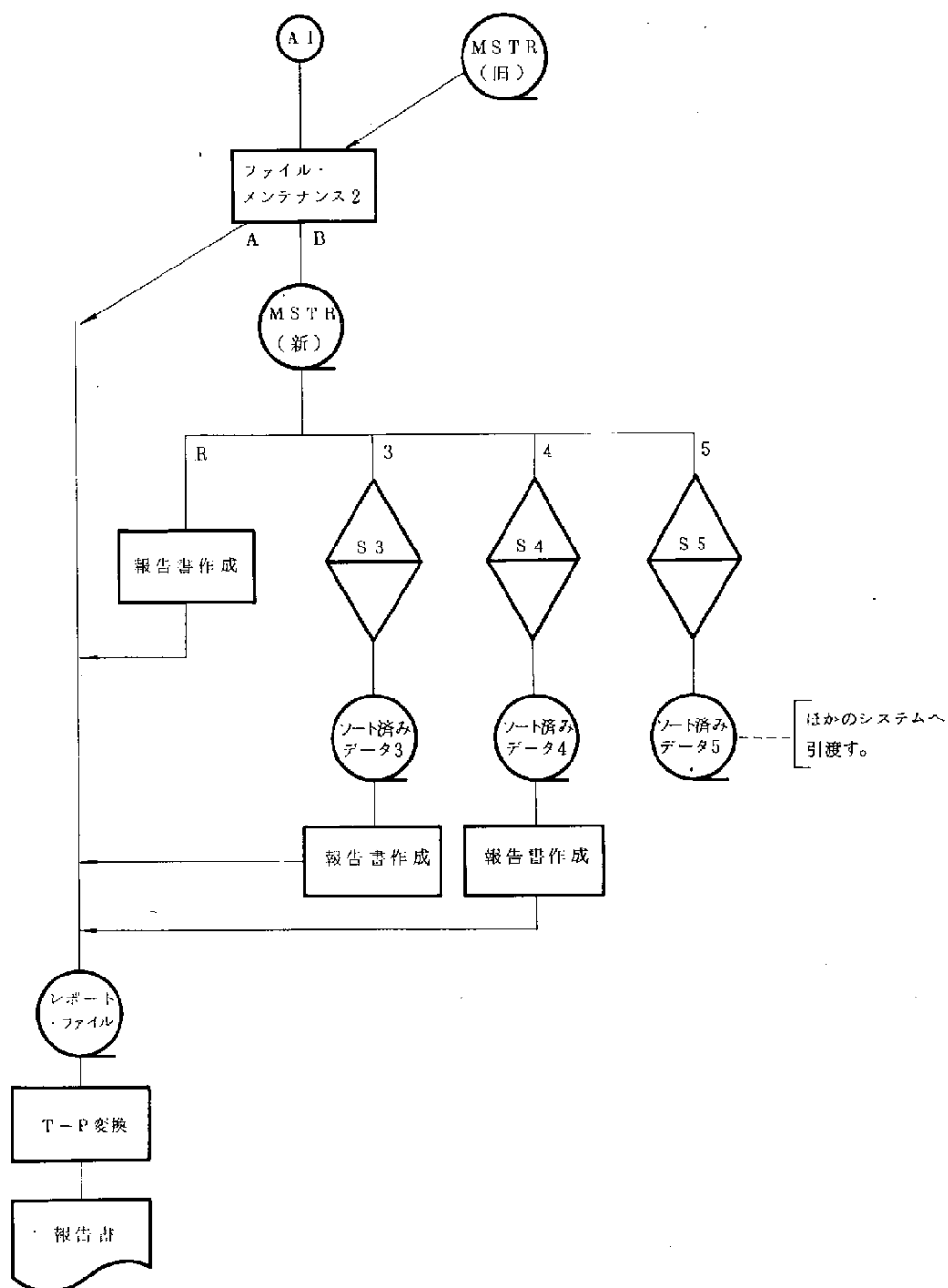


解図 1 1

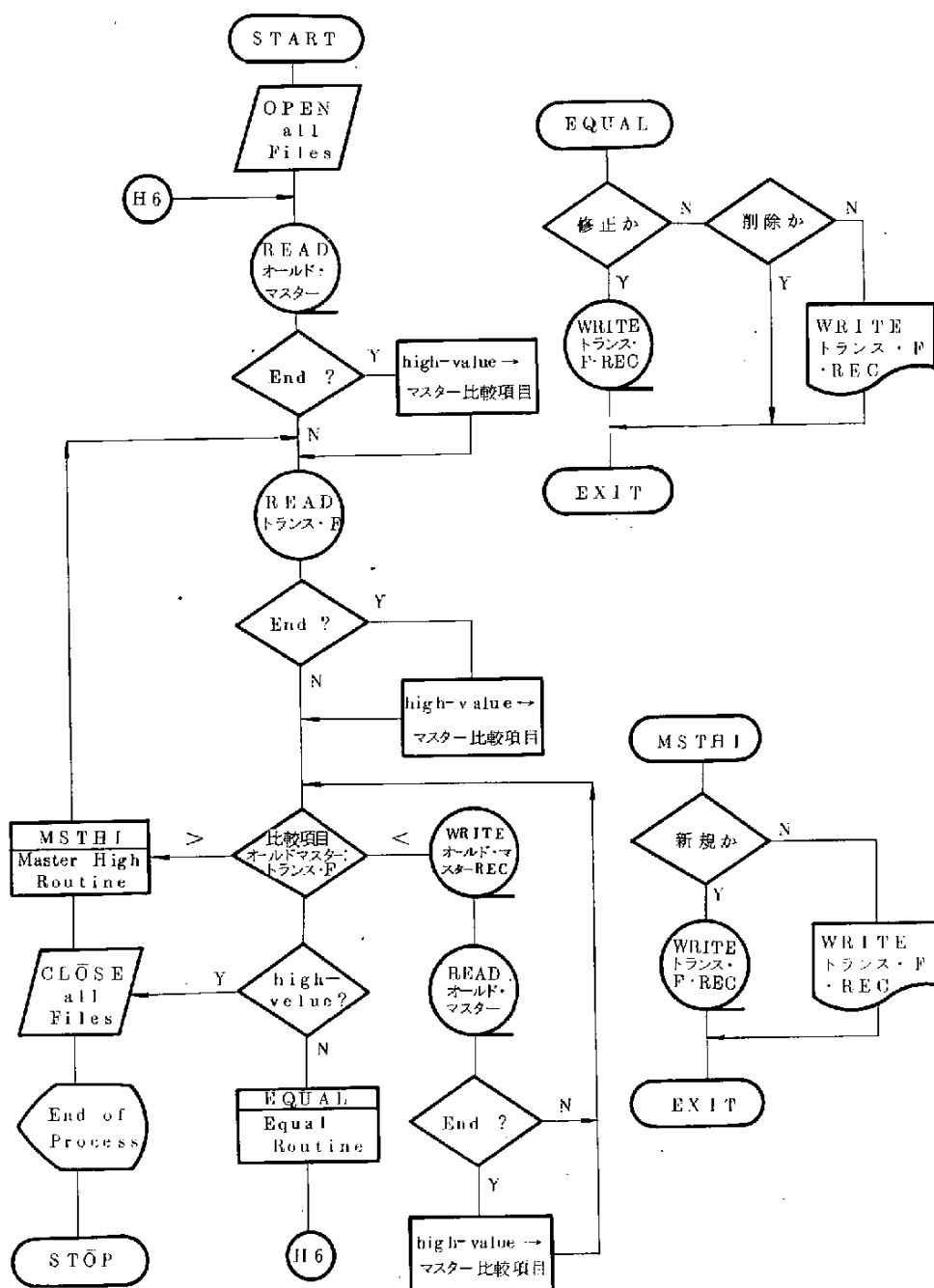
月報作成プロセス流れ図の例

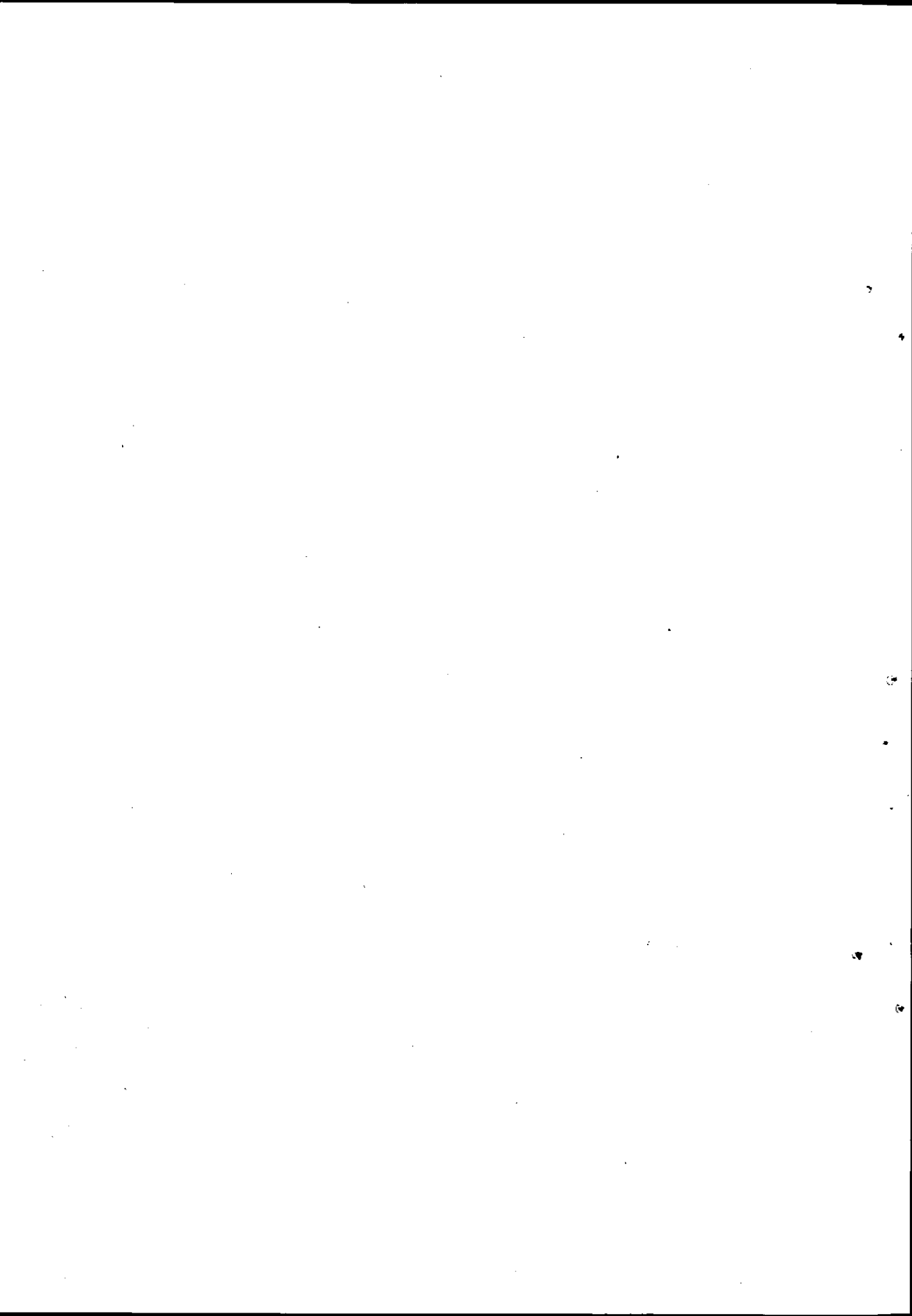


解図11 (つづき)



月報作成プログラム流れ図の例





Ⅳ 電子計算機プログラム用言語

解説（抜粋）

F O R T R A N（水準 3 0 0 0）

電子計算機プログラム用言語

1. 日本工業規格 FORTRAN 制定のいきさつ

(省略)

2. FORTRAN 規格化の目的に関する考察

2.1 序 論

今日広く使用されている電子計算機に対して、“FORTRAN”という名前で非常によく似た一群のプログラム用言語の処理系が作られている。この“FORTRAN”の処理系が広く使われているために、“FORTRAN”は計算機や情報処理組織の共通プログラム言語中で最も普通に使用されるものの1つとなっている。

2.2 FORTRAN の歴史的発展と現状

FORTRAN 語の最初の目的は、つぎのとおりである。

“FORTRAN 語は数値計算のあらゆる問題を表現できることを意図したものである。特に、多くの公式と変数を含む問題を容易に取扱い、各変数は3個までの独立した添字を持つことができる。しかし、機械語が数値的な意味ではなく論理的な意味を持つ場合には、FORTRAN 語は不十分であり、このような問題はまったく表現できないこともある。けれども、FORTRAN 語で直接表現できない論理操作も、ライブラリ・ルーチンを組込めるようにしておくことで可能になる。”この引用は“FORTRAN Automatic Coding System for the IBM 704 EDPM”(1956年10月15日)による。

最初のFORTRANの処理系は1958年に修正され、通常“FORTRAN II”と呼ばれる拡張されたFORTRAN語で書いたプログラムを処理できるようになった。FORTRAN IIの使用は急速にひろがり、その処理系は種々の構造と能力を持つ電子計算機で使えるようになった。

1962年から“FORTRAN IV”の処理系が現れ始め、FORTRAN IIの処理系がかなり広く使用されているにもかかわらず、ますます多く使用されるようになった。FORTRAN IVはFORTRAN語に追加をしたばかりでなく、FORTRAN IIのプログラムをFORTRAN IVの処理系では直接処理することのできないような変更も行なった。そのためFORTRAN IIのプログラムをFORTRAN IVに変換するプログラムが使用されてきた。FORTRAN IIとFORTRAN IVの関係については、2.5で簡単に説明する。

FORTRAN IVの出現によってFORTRAN語族が2分されたことに加えて、時の経過とともに各処理系間で言語上の差異が生じた。そして、これらの差異が処理系間におけるFORTRANプログラムの互換性を制約している。これらの差異は、一般につきのような理由による。

- (1) 言語の適用範囲を拡張するため
- (2) 使用者のために言語を単純化するため
- (3) 計算組織をより効率よく利用するため
- (4) 処理系の動作を単純にし、早めるため
- (5) 明確な規格がないため“FORTRAN”とは何かということに関しての誤解や解釈の相違のため

2.3 目 的

この規格は、FORTRAN語で書かれたプログラムの形式および解釈のしかたを規定し、いろいろなデータ処理組織で使用するプログラムの高度な互換性を図ることを目的とする。処理系がこの規格を満たしているというには、少なくとも、この規格に述べられている形式と諸関係を受入れ、定められたとおりに解釈しなければならない。

この規格で述べられている形式および諸関係の解釈に影響を与えないかぎり、あることを要求しているいかなる文章も「その要求を満たしていなければ、この規格はなんらの解釈をも与えない」という文章によって置きかえることができる。

のみならず、あることを禁止しているいかなる文章も「その禁止事項をおかしたならば、この規格はなんらの解釈をも与えない」という文章によって置きかえることができる。

2.4に与える基準が互換性の水準を決め、3つのFORTRAN規格となった。この水準が、解説の3.で論議されるように、言語の範囲や細部に反映している。

2.4 FORTRAN規格化における基準

FORTRANを規格化するにあたって用いた基準は（ほぼ順位に従って），つぎのとおりである。

- (1) 異種処理系間でのFORTRANプログラムの互換性
- (2) 現在のものと矛盾しないこと
- (3) FORTRAN使用者にとって明確かつ単純であること
- (4) 広範囲の計算機の能率的な動作への適合性
- (5) 言語に将来の発展の余地を残すこと

このFORTRANの規格を制定するにあたっては，現在実用されていない言語機能は新たに付け加えず，もっぱら現行のFORTRANの成文化にとどめた。

FORTRANの過去および現在の広範な使用状況からみて，規格の制定にあたって言語の設計よりむしろ言語の定義に努め，(1)規格の内容が，実用の場で価値を証明されたもののみを反映するようにし，また，(2)この規格を受入れやすいようにした。

2.5 FORTRAN II と FORTRAN IV

（省略）

2.6 広範囲の計算機のためのFORTRAN

互換性の基準と，なるべく広い範囲の機械への適合性の基準は矛盾する。小さな計算機で能率的な動作をするためには，FORTRAN語の比較的使われない部分を省略することが望ましい。しかし，小さな計算機のために言語を制限することは，もっと大きな言語を簡単に取扱える処理系にとって，規格化の利点を否定するものである。

そこで3つの水準を設けることにした。こうすれば利用する計算機に最も適すると思う水準の処理系を作ることができる。下位のFORTRANで書いたプログラムは，上位のFORTRANでも正しいプログラムである。

しかし，3水準が存在することは，上位のFORTRANを受付ける処理系用に書いたプログラムは，下位のFORTRANの処理系では一般には受付けてくれない，という意味で互換性を制約する。各水準の相違点の要約を4.に記す。

3. 本文各節の注

(省略)

4. JIS FORTRANの3水準間の主な相違点

以下に、3つの水準の間の主な相違点を述べる。これはただ便宜のためにあげただけであって、完全な比較のためには各節を調べなければならない。

FORTRAN(水準7000)にあって、FORTRAN(水準5000)にもFORTRAN(水準3000)にもないもの：

- (1) FORTRAN用の文字の中の“通貨記号”
- (2) 複素数型と倍精度実数型のデータ、DまたはGを含む基本欄記述子
- (3) DATA文と初期値設定副プログラム
- (4) 文字型のデータとAを含む基本欄記述子。したがって、書式を実行中に読込むことはできない。

FORTRAN(水準3000)だけが違っているもの、またはFORTRAN(水準3000)だけにあるもの：

- (1) 英字名は最大5文字。FORTRAN(水準7000)、FORTRAN(水準5000)では6文字。
- (2) 論理型のデータ、論理式、関係式、論理IF文およびIを含む欄記述子はない。
- (3) 型宣言文はない。
- (4) EXTERNAL文はない。
- (5) 3次元の配列はない。
- (6) 関数副プログラムで、その引数や共通ブロックのデータを定義したり、再定義したりしてはならない。
- (7) COMMON文に配列宣言子を書くことは許されない。
- (8) 名前付き共通ブロックはない。
- (9) ASSIGN文と割当てGOTO文はない。

- (10) P A U S E 文における 8 進数字は最大 4 個。
- (11) 書式づけられた記録の印刷のときの行送り制御はない。
- (12) F O R M A T 文において
 - (a) けた移動子はない。
 - (b) F 変換による入力 of データに指数があつてはならない。
 - (c) 欄記述群のかつこの中に別の欄記述群が含まれていてはならない。
- (13) 外部関数は、共通ブロックのデータまたは E Q U I V A L E N C E 文によって共通ブロックのデータと結合しているデータの値を変えてはならない。
- (14) すべての D I M E N S I O N 文は、すべての C O M M O N 文の前になければならず、すべての C O M M O N 文は、すべての E Q U I V A L E N C E 文の前になければならない。
- (15) 文の番号は、1 ~ 4 個の数字から成る。
- (16) 整合寸法はない。
- (17) 外部手続きの仮引数が配列名の場合には、それに対応する実引数は配列名でなければならない。

5. 注 意 事 項

5.1 使用者への注意

(1) この規格や個々の処理系の解説書などに記されている必要条件は必ず守り、禁止条件もできるだけ守ることを要望する。禁止条件を犯した場合には、処理系が下記の3つのうちのどれをとるかは不明である。

(a) 何をするか不明。

(b) 警告文が出る。

(c) 偶然にうまくいく。ただし、この場合うまくいったとしても他の処理系でうまくいくと期待してはならない。

5.2 処理系製作者への注意

(1) 処理系の製作者は必要条件についてできるだけ誤りを検出すること。禁止条件については誤りを検出してもしなくてもよい。ただし、誤りを検出して警告文を出すにとどめるようにする。

(2) 1つの文に含まれる継続行の数に関しては、その上限を指定してもかまわない。国際規格の最低水準では、この上限は5個である(本文3.3参照)。

(3) DOの範囲を実行能率のよいプログラムに変換するには、たとえば、

```
DO 1 J=1, N
```

```
1 SP=SP+A(I, J)*B(J, K)
```

を

```
PROD(A, B)=A(I, J)*B(J, K)
```

...

```
DO 1 J=1, N
```

```
1 SP=SP+PROD(A, B)
```

と書かれては困るので、文関数定義式の式中に配列要素名を許さず、また文関数の仮引数に配列名を許さない。したがって、能率に影響を与えないかぎり制限を緩和することは可能であろう。たとえば、添字式がすべて定数の配列要素名を式中に認めてもよからう。

(4) 本文8.1.2の“文関数の引用が実行されると、実引数の値と仮引数との結合が行なわれる”

ということは、文関数の場合の結合の方式を外部手続きの場合と違って、たとえば、文関数では値による結合を行ない、外部関数では名前による結合を行なって、実行の能率をあげてもよいことを示している。

6. プログラム例

(省略)

V 電子計算機プログラム用言語

(抜 粋)

COBOL

Programming Language COBOL
for Computers

電子計算機プログラム用言語

1. 適用範囲 この規格は、COBOL で表現されるプログラムが、いろいろな種類の自動データ処理系で用いられるときに、それらのプログラムの交換可能性をできるだけ高くするための、プログラムの形式とその解釈について規定する。

2. 用語の意味 この規格で用いるおもな用語の意味は、つぎのとおりとする。

(1) 一意名 (identifier) 必要ならば、いくつかの修飾語、添字、指標を構文的に止しくつけたデータ名であって、データ項目を一意に参照する。

(2) 一連項目 (contiguous items) データ部中の隣りあった項で記述された項目であって、互いに直接の階層関係を有するもの。

(3) 印刷集団 (print group) 報告集団と同義。

(4) 英字 (alphabetic character) 以下の文字。

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z および空白

(5) 英数字 (alphanumeric character) 計算機で使える任意の文字。

備考 特殊文字、記号、かな文字などを含んでもよい。

(6) 演算符号 (operational sign) 数字データ項目が正か負かをきめるのにつける算術符号。

(7) カウンタ (counter) 数値または数値の諸表現をたぐわえるデータ項目であって、その数値に対して他の数値を加減したり、変更したり、破算したり、任意の正または負の値を設定したりできる。

(8) 環境句 (environment clause) 環境部の記述項に現われる句。

(9) キー (key) いくつかのデータ項目であって、その内容がいっしょになってレコードの所在やデータの順序づけを指定する。

(10) 記述項 (entry) 句の続いた組であって、終止符と空白でとめる。COBOL の原始プログラムの見出し部、環境部、データ部に書く。

(11) 記述項の主体 (subject of entry) データ部の記述項において、レベル指示語またはレベル番号のすぐうしろに書く作用対象または予約語。

(12) 記述項の述語 (object of entry) データ部の記述項において、記述項の主体のうしろに書く作用対象と予約語などの組。

(13) 基本項目 (elementary item) 論理的にそれ以上は細分されないデータ項目。

(14) 句 (clause) ある記述項の性質を指定するために、COBOL の語などを並べたもの。

(15) 句読文字 (punctuation character) 以下の文字。

文 字	意 味
,	コンマ
;	セミコロン

.	終止符
"	引用符
(	左かっこ
)	右かっこ
	空白

- (16) 形式(format) 一組のデータのある並び方。
- (17) 降順のキー(descending key) データ項目の比較の規則にしたがって、最も高い値のキーから始まり最も低い値のキーにいたる順序にデータが並べられる、その値を有するキー。
- (18) 構成節(configuration section) 環境部中の節であって、翻訳用計算機および実行用計算機の全般的な仕様を書く。
- (19) COBOL 字の組(character set) 以下の51字の集合。

文 字	意 味
0, 1, , 9	数字
A, B, , Z	英字
	空白
+	正号
-	負号またはハイフン
*	星印
/	斜線
=	等号
¥	通貨記号
,	コンマまたは小数点
;	セミコロン
.	終止符または小数点
"	引用符
(	左かっこ
)	右かっこ
>	より大きい記号
<	より小さい記号

- (20) COBOL の原始プログラム(COBOL source program) 問題を解くための処理を、COBOL の言語、形式、文法を用いて表現したもの。
- (21) COBOL の語(COBOL word) 以下の文字だけからなる80字以内の文字列。A, B, C, . . . , Z, 0, . . . , 9, -。ハイフン(-)は、語の最初や最後には書けない。語は、分離符で区切る。単に語(word)ということもある。
- (22) COBOL の実行用プログラム(COBOL object program) COBOL の原始プログラムを翻訳した結果得られる一組の機械語命令。
- (23) 最右端(low order end) 文字の列の右端の文字。
- (24) 最左端(high order end) 文字の列の左端の文字。
- (25) 作業場所節(working-storage section) データ部中の節であって、独立項目やレコードからなる作業場所データ項目を記述する。

- (26) 作成者のきめた名前 (implementor-name) 作成者のきめた語であって、作成者の処理系で用いられる装置や機能をさす。
- (27) 作用対象 (operand) 命令の書き方または記述項の書き方のなかで、日本語で与えられている語。
- (28) 算術演算子 (arithmetic operator) 以下の1字またはきまった2字の組合せ。

文 字	意 味
+	加算
-	減算
*	乗算
/	除算
**	べき乗

- (29) 算術式 (arithmetic expression) 数字基本項目の一意名、数字定数、数字基本項目の一意名や数字定数を算術演算子でつないだもの、算術式を算術演算子でつないだもの、またはかっこでくくった算術式。
- (30) 算術式用文字 (arithmetic expression character) 算術演算子と同義。
- (31) 実行時 (execution time, object time) 実行用プログラムを実行する時期。
- (32) 実行用計算機 (object-computer) 環境部の段落の名前であって、実行用プログラムを実行する計算機の環境をこの段落で記述する。
- (33) 実際のキー (actual key) 大記憶媒体中のレコードの物理的な位置を直接表現するキー。
- (34) 実際の小数点 (actual decimal point) データ項目中の小数点のけた位置を、小数点を表わす終止符(。)またはコンマ(,)を用いて物理的に表現したもの。
- (35) 指標 (index) 計算機の記憶場所またはレジスタであって、その内容は、表のなかの要素を指定するのに用いられる。
- (36) 指標つきデータ名 (indexed data-name) データ名のうしろに、かっこでくくったいくつもの指標名をつけて一意名にしたもの。
- (37) 指標データ項目 (index data item) 指標名に対応する値が、作成者のきめた形式でたぐわえられるデータ項目。
- (38) 指標名 (index-name) 少なくとも1字の英字を含む語であって、ある特定の表に関係づけられた指標を命名する。
- (39) 修飾語 (qualifier) 同じ系列のなかの低いレベルの一意でないデータ名を一意にするデータ名、または一意でない段落名を一意にする節名。
- (40) 修飾されたデータ名 (qualified data-name) データ名のうしろに、連結語 OF または IN と、修飾語のデータ名との組をいくつかつけて一意名にしたもの。
- (41) 集団項目 (group item) 基本項目または集団項目のひとつながりの組に名前をついたもの。
- (42) 出力手続き (output procedure) 分類ずみのレコードを、分類用ファイルから引き取るために実行する一組の命令。
- (43) 順呼出し (sequential access) レコードをファイルから読み込んだり、そこに書き出したりするときの呼出し方式であって、各レコードは、論理的にそれに先行するレコードと後続するレコードとをもつ。ファイルをはじめて参照すると、論理的に先行するレコードをもたないレコードが呼び出される。2番目以降の参照で呼び出されるレコードは、前に呼び出されたレコードに論理的に後続する

レコードである。レコード間のこの先行後続の関係は、レコードをファイルに書き出すときにきめられる。

- (44) 条件(condition) 単純条件、または単純条件と論理演算子との構文的に正しい組合せであって真理値のきめられるもの。
- (45) 条件変数(conditional variable) 条件名をつけられた値をとるデータ項目。
- (46) 条件名(condition-name) 条件変数がとりうる値の組のなかの、特定の値、値の組もしくは値の範囲につけたデータ名、または作成者のきめたスイッチもしくは装置の状態につけた名前。
- (47) 条件名条件(condition-name condition) ある条件変数の値が、その条件名に割り当てられている値の組に含まれているかどうかによって真理値のきまる命題。
- (48) 条件命令(conditional statement) 条件の真理値によって実行用プログラムがつきにとる動作がきまる命令。
- (49) 昇順のキー(ascending key) データ項目の比較の規則にしたがって、最も低い値のキーから始まり最も高い値のキーにいたる順序にデータが並べられる、その値を有するキー。
- (50) 字類条件(class condition) ある項目の内容がすべて英字であるか、すべて数字であるかによって真理値のきまる命題。
- (51) 真理値(truth value) 条件を評価した結果の表現であって、つぎの二つの値のいずれかになる。
成立(真) 不成立(偽)
- (52) スイッチ状態条件(switch-status condition) オンまたはオフに設定できる作成者のきめたスイッチが、所定の状態にあるかどうかによって真理値のきまる命題。
- (53) 数字(numeric character) 以下の文字。0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- (54) 数字項目(numeric item) 0から9までの数字のいくつかからなる文字の列で表わされる値だけを内容とするように記述されたデータ項目。演算符号がつくことも、つかないこともある。
- (55) 数字直定数(numeric literal) いくつかの数字からなる直定数。右端以外の位置に小数点があってもよい。左端に算術符号があってもよい。
- (56) 制御脚書き(control footing) 制御集団の最後の報告集団であって、その制御集団が作られるたびに一つ作られる。
- (57) 制御頭書き(control heading) 制御集団の最初の報告集団であって、その制御集団が作られるたびに一つ作られる。
- (58) 制御集団(control group) 報告書の制御の階層を規定する各制御用データ項目に関係する一組のデータ。ある制御用データ項目に関係する制御集団は、いくつかの制御頭書き報告集団、いくつかの制御脚書き報告集団、関連するいくつかの明細集団の組からなる。
- (59) 制御の階層(control hierarchy) 制御用データ項目の指定された序列。
- (60) 制御の切れ目(control break) 制御の階層を規定するデータ項目の値が変化すること。
- (61) 制御用データ項目(control data item) COBOL の原始プログラムのファイル節または作業場所節に書かれたデータ項目であって、制御の階層中で指定され、その値の変化がみつかる制御の切れ目となる。
- (62) 正書法(reference format) COBOL の原始プログラムを書くときの標準体裁。
- (63) 整数(integer) 想定した小数点の右側にけた位置をもたない数字定数または数字データ項目。一般形式中でこの用語を用いたときは、小数点も符号も含まない数字定数をさす。想定した小数点の右側

にけた位値をもたない数字データ項目を、整数項目とよぶ。

- (64) 正負条件(sign condition) 算術式またはデータ項目の数値が、ゼロより小さいか、ゼロより大きい、またはゼロに等しいかによって真値のきまる命題。
- (65) 節(section) 一つ以上の段落や記述項の組であって、節の見出しで始まる。
- (66) 節の見出し(section header) 語の組合せであって、環境部、データ部、手続き部の各節を識別するために最初におく。環境部とデータ部とは、節の見出しは予約語の列であり、つぎのものがある。

環境部の節の見出し

CONFIGURATION SECTION.

INPUT-OUTPUT SECTION.

データ部の節の見出し

FILE SECTION.

WORKING-STORAGE SECTION.

REPORT SECTION.

手続き部の節の見出しは、節名、予約語SECTION、優先番号⁽¹⁾、終止符からなる。

注⁽¹⁾ 与えないこともある。

- (67) 節名(section-name) 手続き部に書いた節を識別する語。
- (68) 宣言部分(declaratives) 手続き部の最初にあつて、見出しDECLARATIVESと見出しEND DECLARATIVES とにはさまれたいくつかの専用の節。各節は、順次実行の手続き、作成者の入出力システム、あるいは報告書機能の管理のもとに実行される。各節は、節の見出し、翻訳指示のCOPY(複写)文またはUSE(使用)文、およびそれに従属するいくつかの段落の組からなる。
- (69) 想定した小数点(assumed decimal point) データ項目中の小数点位置が、実際の文字を用いなくて指定されたもの。この想定した小数点は、機能的な意味を有するが、物理的に表現された実体はない。
- (70) 添字(subscript) 表のなかの要素をさす値を有する整数。
- (71) 添字つきデータ名(subscripted data-name) データ名のうしろに、かっこでくくったいくつもの添字をつけて一意名にしたもの。
- (72) 大記憶(mass storage) データを順序的にも非順序的にも構成し維持できる媒体。
- (73) 大記憶管理システム(mass storage control system) 大記憶ファイルの処理を管理し、スケジュールを調整する入出力管理システム。
- (74) 大記憶ファイル(mass storage file) 大記憶媒体に収められたレコードの集まり。
- (75) 大記憶ファイル区域(mass storage file segment) 大記憶ファイルの部分であつて、環境部中のFILE-LIMIT(ファイルの範囲)句によって、上限と下限とがきめられる。
- (76) 単項演算子(unary operator) 算術式中で、変数または左かっこの前につける算術演算子“一”であつて、算術式の値を-1倍する効果を有する。
- (77) 単純条件(simple condition) 以下の条件のうちの一つ。

比較条件

字類条件

条件名条件

スイッチ状態条件

正負条件

(条件)

- (78) 段落 (paragraph) 手続き部では、段階名のうしろにいくつかの文を書いたもの。また、見出し部と環境部とでは、段落の見出しのうしろにいくつかの記述項を書いたもの。
- (79) 段落の見出し (paragraph header) ある予約語のうしろに終止符と空白をつけたものであって、見出し部と環境部とのすべての記述項を識別するために最初におく。段落の見出しには、つぎのものがある。

見出し部の段落の見出し

PROGRAM-ID.
AUTHOR.
INSTALLATION.
DATE-WRITTEN.
DATE-COMPILED.
SECURITY.
REMARKS.

環境部の段落の見出し

SOURCE-COMPUTER.
OBJECT-COMPUTER.
SPECIAL-NAMES.
FILE-CONTROL.
I-O-CONTROL.

- (80) 段落名 (paragraph-name) 手続き部の段落を識別するために段落の先頭を書く語。
- (81) 注記 (comment) 原始プログラム中で、見出し部または手続き部に書く注釈。
- (82) 直定数 (literal, literal constant) 文字の列であって、その値はそこに書いてある文字の組によって表わされる。
- (83) データ記述項 (data description entry) データ部中の記述項であって、レベル番号、必要ならデータ名、および必要なデータ句の組からなる。
- (84) データ句 (data clause) データ部のデータ記述項中の句であって、データ項目のある性質を記述する情報を含む。
- (85) データ項目 (data item) 基本項目、一つのレコード中の基本項目の集団に名前をついたもの、またはレコード。
- (86) データ名 (data-name) 少なくとも1字の英字を含む語であって、データ部の記述項を命名する。一般形式中でこの用語を用いたときは、規則でとくに断わってないかぎり、添字や指標をつけてはならないし、また修飾されていてもいけない。
- (87) 手続き (procedure) 手続き部中の一つの段落もしくは節、または機能的に続いたいくつかの段落もしくは節。
- (88) 手続き部の終り (end of procedure division) COBOL の原始プログラムの書き終りであって、それ以降には手続きを書かない物理的な位置。

- (89) 手続き名 (procedure-name) 原始プログラム中の段落または節を参照するのに使う語であって、段落名⁽¹⁾および節名がある。

注⁽¹⁾ 修飾されていてもよい。

- (90) 動詞 (verb) COBOL 翻訳ルーチンまたは実行用プログラムのとる動作を表現する語。
- (91) 登録名 (library-name) 一組の COBOL 記述項や手続きからなる登録項を識別するための語であって、手続き名の構成規則にしたがう。COBOL 登録集を呼び出すときに、登録名のつづりのどの部分が実際に使われるかは作成者がきめる。
- (92) 特殊名 (special-names) 環境部の段落の名前であって、作成者のきめた名前と利用者のつけた呼び名とを、この段落で関係づける。
- (93) 特殊文字 (special character) 以下の文字。

文 字	意 味
+	正号
-	負号
*	星印
/	斜線
=	等号
¥	通貨記号
,	コンマまたは小数点
;	セミコロン
.	終止符または小数点
"	引用符
(	左かっこ
)	右かっこ
>	より大きい記号
<	より小さい記号

- (94) 特殊レジスタ (special register) 翻訳ルーチンが作り出す記憶場所であって、COBOL のある仕様を用いると発生する情報をたぐわえるのが本来の用途である。
- (95) 独立項目 (noncontiguous item) 作業場所節中のデータ項目であって、他のデータ項目と何の階層関係ももたない。
- (96) 入出力管理 (i-o-control) 環境部の段落の名前であって、プログラムが実行時に用いる再運転点、いくつかのデータファイルによる領域の共用、単一の入出力装置上の複数個のファイルをここで指定する。
- (97) 入出力節 (input-output section) 環境部中の節であって、実行用プログラムが利用するファイルと外部媒体に命名し、プログラム実行時のデータ移送と取扱いとに関する情報を指定する。
- (98) 入力手続き (input procedure) レコードを分類用ファイルに引き渡すために実行する一組の命令。
- (99) 比較演算子 (relational operator) 以下に示す予約語、比較文字、予約語の列、または予約語の列と比較文字との組であって、比較条件を書くのに使う。

比較演算子	意 味
IS [NOT] GREATER THAN	より大きいまたは より大きくない。
IS [NOT] >	
IS [NOT] LESS THAN	より小さいまたは より小さくない。
IS [NOT] <	
IS [NOT] EQUAL TO	等しいまたは等し くない。
IS [NOT] =	

(100) 比較条件 (relation condition) ある算術式またはデータ項目の値と、もう一つの算術式またはデータ項目の値とが、所定の比較関係を満たしているかどうかによって真理値のきまる命題。

(101) 比較文字 (relation character) 以下の文字。

文 字	意 味
>	より大きい
<	より小さい
=	等しい

(102) 必要語 (key word) プログラムを書くときに、ある書き方のなかでは必ず書かないといけない予約語。

(103) 表 (table) 論理的につながったデータ項目の組であって、データ部の OCCURS (反復) 句を用いて記述する。

(104) 表意定数 (figurative constant) 数値、文字または文字の列を表わす予約語。

(105) 標準データ形式 (standard data format) COBOL のデータ部でデータの性質を記述するときの概念であって、無限の幅と長さをもった印刷ページ上のデータ形式を想定してデータの性質や特性を表現する。計算機の内部や特定の外部媒体上にデータをたくわえる形式ではない。

(106) 標準文字列 (standard characters) データ項目の内容となる文字列であって、その長さは標準データ形式にしたがって測られる。

(107) 表の要素 (table element) 反復する項目の組が表になったなかのデータ項目。

(108) 部 (division) ある種の規則にしたがって構成されたいくつかの節や段落。COBOL プログラムには、つぎの四つの部がある。

見出し部 (identification division)

環境部 (environment division)

データ部 (data division)

手続き部 (procedure division)

(109) ファイル (file) レコードの集まり。

(110) ファイル管理 (file-control) 環境部の段落の名前であって、原始プログラムで用いるファイルを、この段落で宣言する。

(111) ファイル記述項 (file description entry) データ部のファイル節中の記述項であって、レベル指示語 FD、ファイル名、および必要なファイル句の組からなる。

(112) ファイル句 (file clause) データ部のファイル記述項、分類用ファイル記述項、報告書記述項に現われる句。

(113) ファイル節 (file section) データ部中の節であって、ファイル記述項を書く。

(114) ファイルの範囲 (file-limit) 大記憶ファイルを構成する各区域の上限と下限との組であって、

大記憶媒体の物理的限界位置の内側に収められている。

- (115) ファイル名(file-name) 少なくとも1字の英字を含む語であって、データ部で記述されているファイルを命名する。
- (116) 部の見出し(division header) それぞれの部の始まりを明示するCOBOLの語の列。以下のものがある。

IDENTIFICATION DIVISION.

ENVIRONMENT DIVISION.

DATA DIVISION.

PROCEDURE DIVISION.

- (117) プログラム名(program-name) COBOLの原始プログラムを識別するための語。
- (118) ブロック(block, physical record) 計算機で入出力装置に記憶させるのにつぎのよいデータの物理的な単位。物理レコード(physical record)と同義である。ブロックは、ふつうはいくつかのレコードまたはその部分からなる。ブロックの大きさは、そのファイルの大きさや、ブロックに含まれたり重なったりしているレコードの大きさなどと直接の関係はない。
- (119) 文(sentence) いくつかの命令の列であって、終止符と空白でとめられたもの。
- (120) 分離符(separator) 読みやすくするためにつける文字。
- (121) 分類用ファイル(sort file) SORT(分類)命令で分類されるレコードの集まり。この分類用ファイルは、分類機能によってのみ作り出され、利用される。
- (122) 分類用ファイル記述項(sort file description entry) データ部のファイル節中の記述項であって、レベル指示語SD、ファイル名、および必要なファイル句の組からなる。
- (123) ページ(page) ひとつながりの報告書データを物理的に区分する横の単位であって、その区分は、報告書の体裁や報告書を出力する外部装置の性質にもとづく。
- (124) ページ脚書き(page footing) 報告書のページの最後の報告集団であって、ページ送り条件をみつけた結果、ページが送られる直前に作られる。
- (125) ページ頭書き(page heading) 報告書のページの最初の報告集団であって、ページ送り条件をみつけた結果、ページを送った直後に作られる。
- (126) 編集用文字(editing character) 以下の1字またはきまった2字の組合せ。

文 字	意 味
B	空白
0	ゼロ
+	正
-	負
CR	貸方(credit)
DB	借方(debit)
Z	ゼロ抑制
*	小切手改変防止
¥	通貨記号
,	コンマまたは小数点
.	小数点

- (127) 変数(variable) プログラムの実行によって値が変更されるデータ項目。算術式中の変数は、数字基本項目でなければならない。
- (128) 報告集団(report group) 報告書中で互いに関係をもつ一組のデータ。
- (129) 報告書(report) 報告書ファイル中に記述された一組のデータを表示したもの。
- (130) 報告書脚書き(report footing) 報告書の最後の報告集団であって、報告書が終了するときただ1回だけ作られる。
- (131) 報告書頭書き(report heading) 報告書の最初の報告集団であって、報告書が開始されるときにただ1回だけ作られる。
- (132) 報告書記述項(report description entry) データ部の報告書節中の記述項であって、レベル指示語 RD、その報告書につけたデータ名、および必要なファイル句の組からなる。
- (133) 報告書節(report section) データ部中の節であって、いくつかの報告書記述項を書く。
- (134) 報告書の行(report line) ページを横割りにした単位であって、文字のつながりである。
- (135) 報告書の体裁(report format) 報告書のページの体裁であって、報告書節で指定する。
- (136) 報告書ファイル(report file) 報告書機能が作り出したレコードの集まりであって、その内容と形式とが、報告書を作成するのに用いられる。
- (137) 報告書名(report-name) 報告書につけたデータ名。
- (138) 補助語(optional word) 言語を読みやすくするための、ある書き方のなかに書く予約語であって、プログラムを書くときに、書くかどうかは利用者にまかされる。
- (139) 翻訳時(compile time) COBOL の原始プログラムを、COBOL 翻訳ルーチンによって実行用プログラムに翻訳する時期。
- (140) 翻訳指示命令(compiler directing statement) 翻訳指示動詞で始まる命令であって、翻訳ルーチンに指示した動作をとらせる。
- (141) 翻訳用計算機(source-computer) 環境部中の段落の名前であって、原始プログラムを翻訳する計算機の環境をこの段落で記述する。
- (142) 無効なキーの条件(invalid key condition) 大記憶ファイルに対して用いる実際のキーの実行時の値が、呼んでいるファイルの範囲の外側をさしているときの条件。
- (143) 無条件命令(imperative statement) 無条件動詞で始まり、無条件にきまった動作をとるように指示する命令。一つの無条件命令が、いくつかの無条件命令のつながったものであってもよい。
- (144) 命令(statement) 手続き部において、動詞に続けて語や記号の構文的に正しい組合せを書いたもの。
- (145) 文字(character) COBOL 語の、それ以上分割できない基本単位。
- (146) 文字位置(column) 報告書の行のなかの、それぞれの文字の位置。けた位置ともいう。
- (147) 文字項目(non-numeric item) 計算機の文字の組中の任意の組合せを内容とするデータ項目。文字項目の項類によっては、使用できる文字が制限される場合がある。
- (148) 文字直定数(non-numeric literal) 引用符でくくった文字の列。計算機の文字の組のうち、引用符以外の文字は何がはいてもよい。
- (149) 文字の大小順序(collating sequence) 計算機が扱える文字を大小比較したときの順位であって、作成者がきめる。

- (150) 文字列 (character-string) ひとつながりの文字の列であって、直定数、語、データ部中の PICTURE (形式) 句の文字列、手続き部中の NOTE (注記) 文の文字列の 4 種類がある。
- (151) 優先番号 (priority-number) 0 から 99 までの値をもつ番号であって、手続き部中の各節に書き、これにもとずいて実行用プログラムが区分化される。
- (152) ユニット (unit) 大記憶の一つのモジュールであって、その大きさは作成者がきめる。
- (153) 呼び名 (mnemonic-name) プログラマの作った語であって、環境部において作成者のきめた名前に関係づけられる。
- (154) 予約語 (reserved word) COBOL の原始プログラム中で使われるきまった語であって、利用者のきめた語として使ってはならない。予約語の表は、4.9 による。
- (155) 乱呼出し (random access) 指定されたレコードを大記憶ファイルから読み込んだり、そこに書き出ししたりするときの呼出し方式であって、作成者の大記憶管理システムの管理のもとに、非順序的な方法でレコードが呼び出される。
- (156) レコード (record, logical record) いちばん包括的なデータ項目。
- (157) レコード記述 (record description) あるレコードに関するいくつかのデータ記述項の全体。
- (158) レコード名 (record-name) レコードにつけたデータ名。
- (159) レベル指示語 (level indicator) きまった 2 字の英字の組合せであって、ファイルの種類または階層中の序列の一つを指示する。
- (160) レベル番号 (level number) 2 けたの数字であって、01 から 49 までの場合にはレコード中の階層構造を指定し、66, 77, 88 の場合にはデータ記述項の特別の性質を指定する。
- (161) 連結語 (connective) 語または句読文字であって、用途はつぎのとおりである。
- (a) データ名や段落名をその修飾語に結びつける。
 - (b) 並べて書いた複数個の作用対象をつなぐ。
 - (c) 論理連結語として条件を作る。
- (162) 論理演算子 (logical operator) 予約語の AND, OR, NOT。条件を書くのに AND と OR を論理連結語として使える。論理否定として NOT を使える。

3. 言語の構造 プログラムには、見出し部、環境部、データ部および手続き部を書く。それぞれの部を構成する言語要素 (language element) は、つぎに示す 8 個の単位のうちの一つ以上に属するものとする。

- (1) 中核 (nucleus)
- (2) 表操作 (table handling)
- (3) 順呼出し (sequential access)
- (4) 乱呼出し (random access)
- (5) 分類 (sort)
- (6) 報告書作成 (report writer)
- (7) 区分化 (segmentation)
- (8) 登録集 (library)

上記の 8 個の単位のうち、中核以外のものを、それぞれ機能単位 (functional processing module) とよぶ。

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

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

3.3 水準の内容 中核および機能単位のそれぞれの水準の技術的な内容は、5.~21. による。

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

3.5 COBOL の作成 ある COBOL の言語仕様を定め、それにもとずいて処理系を作り、利用者に提供する者を、作成者 (implementor) とよぶ。

COBOL を作成するときに、それが、中核と機能単位との各水準を、この規格で規定されたように含んでいれば、その COBOL は、ここに定める COBOL の作成上の要求を満たしているものとみなす。

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

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

注⁽¹⁾ 空集合の水準を除けば、中核、表操作、順呼出しの低い水準からなる。

(2) 最高水準 (maximum JIS COBOL) は、中核と機能単位とのそれぞれの最高の水準を組み合わせたものとする。

3.5.1 ハードウェアとの関係 以下に示す言語要素のうちで、ハードウェアの構成によって作成が左右されるものについては、その COBOL を動かすのに必要なハードウェアの最小構成を、作成者が指定しなければならない。ある COBOL の作成にあたって、これらの言語要素が除外されていたとしても、そのような作成が規格に合致しない (nonstandard) ものとみなされるわけではない。

CLOSE REEL (リールの閉じ)

CLOSE UNIT (ユニットの閉じ)

CLOSE (閉じる) … NO REWIND (巻戻さず) の場合

OPEN (開く) … NO REWIND (巻戻さず) の場合

OPEN I-O (入出力両用フェイルを開く)

OPEN REVERSED (逆読み用に開く)

USE I-O (入出力両用ファイルに使用)

MULTIPLE REEL (多重リール)

MULTIPLE UNIT (多重ユニット)

WRITE (書出し) … ADVANCING 呼び名(呼び名までの行送り)の場合

WRITE (書出し) … INVALID KEY (無効なキー)の場合

スイッチ状態

USE (使用) … REEL LABEL (リールラベル)の場合

USE (使用) … UNIT LABEL (ユニットラベル)の場合

ACCESS MODE IS SEQUENTIAL (呼出し方式は順)

PROCESSING MODE IS SEQUENTIAL (処理方式は順)

ある言語要素で指定するのと同じ機能が、ほかの手段で実現できるようになっている場合には、利用者がそのような言語要素をCOBOLの原始プログラムに書く必要はない。

翻訳ルーチンは、COBOLの原始プログラム中に上記のような言語要素が現われたときに、それを処理できるように作っておかなければならない。

この言語要素が機械語コードを生成するものではない場合に、この機能を実現するためのかわりの表現は、COBOLの原始プログラム中では必要はない。

3.5.2 作成者による規定 以下に示す言語要素は、規格の規定の範囲内でさらに詳細な規定を、作成者が明示しなければならない。

(1) 構文規則の一部を、作成者が定めなければならない言語要素は、つぎのとおりとする。

ASSIGN (割当て)

SOURCE-COMPUTER (翻訳用計算機)

OBJECT-COMPUTER (実行用計算機)

計算機名

記憶容量

スイッチ名

作成者のきめた名前

FILE-LIMIT (ファイルの範囲)

RERUN (再運転)

VALUE OF (ラベル項目の値)

ACCEPT (小入力) … 1回に転送できるデータの最大量

DISPLAY (表示) … 1回に転送できるデータの最大量

ENTER (導入)

USE LABEL PROCEDURE (ラベル手続に使用)

登録名

R欄

(2) 効果の一部を、作成者が定めなければならない言語要素は、つぎのとおりとする。

FILE-LIMIT (ファイルの範囲)

ACTUAL KEY (実際のキー)

SYNCHRONIZED (けたづめ)

COMPUTATIONAL (計算用)

INDEX (指標用)

REDEFINES (再定義) …記憶装置の語にたいする異なったデータ記述を含む場合

USAGE IS DISPLAY (表示用) …演算符号を含む場合

文字項目の比較

ACCEPT (小入力) …1回に転送できるデータの最大量

DISPLAY (表示) …1回に転送できるデータの最大量

算術式

USE ERROR PROCEDURE (誤り手続きに使用)

STOP RUN (実行停止) …開いたままのファイルがある場合

3.5.3 余分な言語要素 COBOL の規格中で規定されていない言語要素や機能が、あるCOBOL の作成にあたって含まれた場合に、それが規格に合致しないものとみなされるわけではない。この場合に、作成者が予約語 (reserved word) の表を拡張し、その結果、規格に合致しているある種の原始プログラムを正しく翻訳できなくなったとしても、さしつかえはない。

規格に合致しているある種の原始プログラムが、文字の置換えによって、正しく翻訳できなくなる場合については、4.3.1の規定による。

4. 言語の全般規定

4.1 序説 COBOL の最高水準に適用される言語の概念と規則とを、以下に規定する。ある機能単位のある水準に、これらの概念や規則のすべてがそのまま適用されない場合には、その水準を規定する簡条で制限事項を述べる。ある機能単位に加えられた制限が、他の機能単位に影響をおよぼすこともある。

例： 中核の水準1は、連結語の使用を禁止している。したがって、中核の水準1と組み合わせられる他の機能単位にも、この制限が適用される。

4.2 一般形式と規則との記法

4.2.1 一般形式 句や命令の要素の特定の並べ方を一般形式 (general format) という。句や命令は、以下に定義する要素からなる。句や命令は、一般形式および規則によって規定する。並べ方が複数通りあるときは、番号をつけて分けて示す。句は、一般形式で示してある順序で書かなければならない。書いても書かなくてもよい句も、それを用いるときは、示された順序で書かなければならない。書き方で示してある順序以外の順序で句を書いてもよいときは、その書き方についての規則中で明記する。必要条件、制限などは、規則として述べる。

4.2.1.1 要素 句や命令を作る要素は、大文字の語、日本語の語、レベル番号、角かっこ、中かっこ、連結語および特殊文字からなる。

4.2.1.2 語 下線のある大文字の語は必要語 (key word) で、その機能を用いるときは必ず書く。下線のない大文字の語は補助語 (optional word) で、原始プログラムに書いても書かなくてもよい。大文字の語は、下線があってもなくてもそのとおりに正しくつづる。

一般形式中の日本語の語は、利用者の与える COBOL の語を代表するのに用いる。一般形式に現われるこれらの日本語の語は、実際のプログラムでは COBOL の語で置き換える。ただし、以下に列記するものは例外であって、それぞれの定義にしたがった COBOL の語の列などで置き換える。その規定は、それぞれの簡条による。

- (1) 命令 (4.7.2 参照)
- (2) 無条件命令 (4.7.2.3.1 参照)
- (3) 算術式 (6.5.1 参照)
- (4) 文字列 (5.5 参照)
- (5) 注記項 (5.3 参照)
- (6) 条件 (5.6.1 参照)
- (7) 直定数 (4.3.2.2 参照)

一般形式のなかに、代表する語がいくつもあるときは、番号や文字をつけて区別する。

4.2.1.3 レベル番号 データ記述項の一般形式に書いてある特定のレベル番号は、COBOL プログラムでその記述項を用いるときには、必ず書かなければならない。

4.2.1.4 角かっここと中かっこ 一般形式で角かっこ〔 〕で囲んである部分は、書くか省くかを利用者が選択する。一般形式で中かっこ { } で囲んである部分は、そのうちの書き方の一つを利用者が選択する。いずれの場合も、選択できるのは縦に並んだ行のうちの一つとする。角かっこや中かっこのなかにただ1行しか書いてないときには、それは反復記号が適用される部分を示す。

4.2.1.5 反復記号 本文中の省略記号“...”は、一部分を省略したことを示す。その意味は文脈による。

一般形式中の反復記号は、利用者の書き方の反復を示す。書き方のうちで反復できるのは、つぎの部分とする。句や命令の書き方で、反復記号“...”の直前左の閉じかっこ〕や}と、それに対応する開きかっこ〔や{とで囲まれた部分の語の組を、その反復記号のところで反復できる。

4.2.1.6 書き方の句読法 書き方では、コンマとセミコロンを句読文字として示す。ただし、記述項や段落の最初の句の直前にコンマやセミコロンを書いてはならない。各部での句読文字の用法は、つぎのとおりとする。

- (1) 見出し部 この部の書き方では、陽には書いてないが、注記項にはコンマとセミコロンを使ってよい。段落そのものは、終止符と空白で終わる。
- (2) 環境部 書き方のコンマとセミコロンは、書いてもよいし省いてもよい。段落そのものは、終止符と空白で終わる。
- (3) データ部 書き方のコンマとセミコロンは、書いてもよいし省いてもよい。記述項そのものは、終止符と空白で終わる。
- (4) 手続き部 添字づけや指標づけのためのかっこのなかのコンマは、必ず書かなければならない。それ以外の書き方のコンマは、書いてもよいし省いてもよい。命令と命令との間には、セミコロンを書いてよい。

参考 COBOL 1969 では、添字づけや指標づけのためのコンマは、書いてもよいし省いてもよい。

4.2.1.7 書き方での特殊文字の用法 書き方に現われる文字“+”、“-”、“>”、“<”、“=”などには下線はないが、その書き方を用いるときには必ず書かなければならない。

4.3 言語の概念

4.3.1 文字の組 COBOL 字の組は、2.(19)に規定する51字とする。

4.3.1.1 文字の類 語のための文字の組は、2.(21)による。

句読点のための文字の組は、2.(15)による。

算術演算子のための文字の組は、2.(28)による。

比較演算子のための特殊文字は、2.(101)による。

編集のための文字の組は、2.(126)による。

COBOL で使う文字は、英字、数字および特殊文字とよばれる文字で、算術式、比較、編集に用いられるものを含む。ある計算機の文字の組が、この文字をすべて含むとは限らないから、他の文字で書き換えてもよい。文字の組が51字より少ないときには、ある1文字を2文字で書き換える。

4.3.1.2 分離符 引用符で囲まれたなか以外の場所にある空白や句読文字は、分離符(separator)とする。空白を使うときは二つ以上書いてよいが、4.で述べる制限は例外とする。

文字列は、その右側を空白、終止符、右かっこ、コンマまたはセミコロンで区切る。

文字列に関連する句読文字の用法は、つぎのとおりとする。

- (1) 文字列を区切るのに終止符、コンマまたはセミコロンを用いたときには、これらのあとに空白を続けなければならない。
- (2) 左かっこの直後や右かっこの直前に空白があってはならない。

4.3.2 文字列 文字列(character-string)とは、ひとつながりの文字の列であって、直定数、語、PICTURE(形式)句の文字列またはNOTE(注記)文の文字列を形成する。

4.3.2.1 語

4.3.2.1.1 語の定義 語(word)とは30字以内の文字の列である。各文字は、つぎの文字の組から選ぶ。A, B, C, ..., Z, 0, ..., 9, -。ただし、語頭と語尾は、ハイフン(-)であってはならない。

4.3.2.1.2 語の種類

4.3.2.1.2.1 データ名 データ名 (data-name) は、少なくとも1字の英字を含む語であって、データ部の記述項を命名する。

4.3.2.1.2.2 条件名 条件名 (condition-name) とは、少なくとも1字の英字を含む語であって、データ項目がとりうる値の組のなかの特定の値、値の組または値の範囲につけた名前である。このデータ項目自体は、条件変数 (conditional variable) という。各条件名は、一意であるかあるいは修飾によって一意にできるものでなければならない。条件変数を、その条件名のいずれかの修飾語として用いてよい。条件変数を参照するのに指標、添字、修飾が必要なときは、その条件名を参照するにも同じ組合せの指標、添字、修飾を用いなければならない。

条件名は、データ部で記述するほかに、環境部の特殊名段落で定義することができ、条件名で、作成者のきめた装置のオンオフの状態に命名する。

条件名は、比較条件の省略した書き方として条件のなかで用いられ、その条件変数が、その条件名に割り当てられている値のうちの一つに等しいかどうかを表わす。

4.3.2.1.2.3 手続き名 手続き名 (procedure-name) は、語であって、手続き部の段落や節を命名する。0～9の数字だけからなる手続き名は、同じけた数で同じ値をもつときに限り互いに等しい。

4.3.2.1.2.4 表意定数 ある種の定数は、表意定数 (figurative constant) といい、きまったデータ名をつける。そのデータ名を表意定数として用いるとき、それを引用符で囲んではならない。表意定数の単数形と複数形は同じことで、どちらを使ってもよい。

きまったデータ名とその意味は、つぎのとおりとする。

- (1) ZERO 文脈により、値ゼロ (0) または何けたかの文字ゼロ (0) を表わす。
ZEROS
ZEROES
- (2) SPACE 何けたかの空白を表わす。
SPACES
- (3) HIGH-VALUE その計算機の文字の大小順序で、最高の値を有する文字の何けたかを
HIGH-VALUES 表わす。
- (4) LOW-VALUE その計算機の文字の大小順序で、最低の値を有する文字の何けたかを
LOW-VALUES 表わす。
- (5) QUOTE 引用符 ("), または引用符のない計算機では、それに相当する文字
QUOTES の何けたかを表わす。プログラム中で、文字直定数を囲む引用符の意味にこの語 QUOTE を用いることはできない。
- (6) ALL 定数 この定数を構成する文字列の何回かの反復を表わす。定数は、文字直定数が "ALL 定数" 以外の表意定数かである。表意定数を用いるときは、語 ALL は、ただ読みやすさのためにだけ用いる。

表意定数が何けたかの文字列を表わすとき、この文字列の長さは、翻訳ルーチンがつぎの規則にしたがい文脈からきめる。

- (1) 表意定数が他のデータ項目と関係づけられたとき、つまり表意定数が他の項目に転記または比較されたときは、表意定数できめた文字は、1字ずつ右のほうに反復され、できあがった列の長さは、関係づけられたデータ項目の文字列の長さと同しくなる。

(2) 表意定数が他の項目と関係づけられないとき、つまり表意定数がDISPLAY (表示) ,

EXAMINE (けた調べ) , STOP (停止) 命令で用いられたときは、列の長さは1字である。

表意定数のうち“ALL 定数”は、DISPLAY, EXAMINE, STOP 命令には使えない。

書き方で定数 (literal) とあるところは、どこでも表意定数を用いてよい。ただし、定数が数字に限定されているときは、表意定数のうちZERO, ZEROS, ZEROES だけが使える。

4.3.2.1.2.5 特殊レジスタ

(1) TALLY TALLY (計数器) という語は、特殊レジスタの名前で、それは暗に符号のない

5けたの整数として記述されており、そのUSAGE (用途) は計算用とする。このTALLY レジスタの本来の用途は、EXAMINE (けた調べ) 命令で発生する情報を保持することである。

このTALLY という語は、整数値の基本データ項目の使えるところなら、どこでもデータ名として使ってよい。

4.3.2.1.2.6 呼び名 呼び名 (mnemonic-name) は、作成者のきめた名前 (implementor-name) を問題向きの名前に、またハードウェアのスイッチの状態を条件名に関連づけるために用いる。

4.3.2.1.2.7 予約語 予約語 (reserved word) とは、COBOL の原始プログラム中で用いる語の特定の組であって、利用者のきめた語としてプログラム中で使ってはならない。

予約語は、つぎの3種類とする。

(1) 必要語 必要語 (key word) とは、プログラムを書くときに、ある書き方のなかでは必ず書かなければならない語である。それぞれの一般形式では、この語は大文字に下線をしてある。必要語は、つぎの3種類とする。

(a) ADD (加算) , READ (読込み) , ENTER (導入) などの動詞。

(b) 命令や記述項の書き方に現われる必要な語。

(c) NEGATIVE (負) , SECTION (節) , TALLY (計数器) などの機能的な用途を有する語。

(2) 補助語 それぞれの一般形式で、下線のしていない大文字の語は補助語 (optional word) といい、利用者は書いても書かなくてもよい。ある書き方のなかの補助語のそれぞれは、書いても書かなくても翻訳ルーチンの翻訳結果にかわりがない。補助語のつづりを変えたり、他の語をかわりに書いたりしてはならない。

備考 補助語を書いたときと書かないときとで、翻訳結果の異なる場合がある。

例： 比較条件の略記法 (6.5.2.1.1 参照) を用いると、

$a > b \text{ AND IS NOT } > c \text{ OR } d$

は、つぎのものと同じである。

$(a > b) \text{ AND } (a \text{ IS NOT } > c) \text{ OR } (a \text{ IS NOT } > d)$

補助語ISを書かない場合の

$a > b \text{ AND NOT } > c \text{ OR } d$

は、つぎのものと同じであり、ISを出した場合とはその意味が異なる。

$(a > b) \text{ AND NOT } (a > c) \text{ OR } (a > d)$

(3) 連結語 連結語 (connectives) は、つぎの3種類とする。

(a) 修飾語の連結語で、データ名や役格名をその修飾語に関係づけるのに用いるOFとIN。

(b) 並べて書いた複数語の作用対象をつなぐ接続子のコンマ(,)。

(c) 条件を作るのに用いる論理連結語。

AND, OR, AND NOT, OR NOT

4.3.2.2 直定数 直定数 (literal) は、文字の列であって、その値は、直定数を構成する文字の順序づけられた組そのものによって示される。直定数は、数字または文字のどちらかの型に属する。

(1) 文字直定数 (non-numeric literal) とは、計算機の文字の組中の引用符を除く任意の可能な文字の列を、引用符で囲んだものである。作成者は、1 字以上少なくとも 120 字までの長さの直定数を許さなければならない。文字直定数の値は、引用符をはずした文字の列そのものである。引用符で囲んだなかの空白は、直定数の部分であり、値の一部に含まれる。文字直定数の項類 (category) は、英数字 (alpha-numeric) とする。

(2) 数字直定数 (numeric literal) とは、0～9 の数字、正号、負号、小数点からなる文字の列である。作成者は、数字の個数が 1 けた以上 18 けたまでの長さの数字直定数を許さなければならない。数字直定数の値は、その文字で表現される数値である。数字直定数の項類は、数字 (numeric) とする。数字直定数を作る規則は、つぎのとおりとする。

(a) 少なくとも 1 字の数字を含む。

(b) 符号文字は、桁数個は書けない。符号を書くときは、直定数の左端の文字位置におく。符号がないとき、その直定数は正とみなす。

備考 直定数が、数字直定数の書き方にあっても引用符で囲んであれば、それは文字直定数であり、翻訳ルーチンでもそのように扱われる。

4.3.2.3 PICTURE 句の文字列 PICTURE (形式) 句の文字列は、COBOL 字の組中の、記号として用いられる文字のある組合せからなる。許される組合せは 5.5.7 による。

4.3.2.4 NOTE 文の文字列 NOTE (注記) 文の文字列は、計算機の文字の組中の文字の任意の組合せからなる。

4.3.3 機種によらないデータ記述の概念 データをできるだけ機種によらないものにするため、データの性質や特性は装置向きの形式ではなく、標準データ形式にあわせて記述する。この標準データ形式は、一般のデータ処理応用に向いている。十進数字は、計算機の基数系によらず、数値を表現するのに用いる。その他の文字は、文字データ項目の値を表現するのに用いる。

4.3.3.1 レコードとファイルの概念 ファイル情報を定義するにあたって、ファイルの物理的な事項とファイルに含まれるデータの論理的性質とを区別する。

4.3.3.1.1 ファイルの物理的な事項 ファイルの物理的な事項は、入出力媒体上の表現にあわせてデータを記述し、つぎの仕様を含む。

(1) ファイル媒体の物理的制限にあわせたレコードのブロック化。

(2) ファイルを識別する方法。

4.3.3.1.2 ファイルの論理的性質 ファイルの論理的性質は、ファイル自体のなかの論理的実体を定義することによってあきらかになる。COBOL プログラムの入出力命令は、一つの論理レコードを参照する。

物理レコード (ブロック) と論理レコード (レコード) を区別する。COBOL のレコードは、関連ある情報の集団であって、一意に指定でき、単位として扱われる。

ブロックは、情報の物理的な単位で、その長さや記録様式は、それぞれの計算機の入出力装置がデータを記憶するのにつごうよくできている。ブロックの長さは、ハードウェアに依存し、装置に含まれる情報ファイルの大きさと直接の関係はない。

一つのレコードが、一つの物理的な単位（ブロック）に収められることもある。また、いくつかのレコードが、一つの物理的な単位に収められることもある。一つのレコードを収めるのに複数個の物理的な単位を必要とすることもある。レコードと物理的な単位との関係を記述するのに、言語の書き方がいくつかある。この関係がきまると、物理的な単位に関係づけられたレコードの呼出し方の管理は、実行用プログラムがひきうける。とくに“物理レコード”というのでなければ、論理レコードの意味でレコードという用語を用いる。

レコードの概念は、ファイルのデータだけでなく、作業場所や定数にも適用する。そこで、作業場所や定数もレコードとして集団化され、一連のデータ記述項を書いてきめる。

4.3.3.1.3 レコードの概念 レコード記述は、あるレコードの性質を記述するデータ記述項の組からなる。データ記述項は、それぞれ、レベル番号、必要ならばデータ名、必要な別々の句の組からなる。

4.3.3.2 レベルの概念 レコードは、レベル構造の概念にしたがって構成される。この概念は、データを参照するためにレコードを細分する必要からくる。いったん細分したものをさらに細分して、もっとこまかいデータを参照できる。

レコードで最も基本的な細分、つまりそれ以上分割できない部分を基本項目（elementary item）という。したがって、レコードは、基本項目のつらなったものであるか、またはレコードそのものが一つの基本項目であるとする。

一組の基本項目を参照するのに、それらをまとめて集団にする。各集団は、いくつかの基本項目の集団に名前をつけたものである。集団はさらにいくつかの集団の集団にまとめられる。したがって、ある基本項目は、いくつかの集団の下位に属する。

4.3.3.2.1 レベル番号 レベル番号（level-number）を用いて、基本項目と集団項目との構成をあきらかにする。レコードはデータ項目のうち最も包括的なものであるから、レコードのレベル番号は1または01で始める。下位のデータ項目には、連続番号とは限らないより大きい49までのレベル番号を割り当てる。特別のレベル番号として、66、77、88があり、この用法は以下に規定する。プログラムを書くとき、各レベル番号を用いるごとに別々の記述項を書く。

ある集団は、それに続く集団項目（group item）か基本項目で、その集団のレベル番号に等しいかより小さいものが出てくるまでのすべてを包括する。ある集団項目の最後の基本項目のすぐつぎに書く項目のレベル番号は、それが基本項目であれ集団項目であれ、先の基本項目の属する集団のうちの一つのレベル番号と同じでなければならない。

レベルの概念が実際には適用できない記述項は、つぎの3種類とする。

- (1) RENAMES（再命名）句で作った基本項目または集団項目の記述項。RENAMES 句でデータ項目を再編成するための記述項には、特別のレベル番号66を割り当てる。
- (2) 作業場所の独立項目。作業場所の独立項目で、他の項目の細分ではなく、またそれ自体細分されていないものの記述項には、特別のレベル番号77を割り当てる。
- (3) 条件名を指定する記述項。条件変数のある値に関係づけられた条件名を指定する記述項には、特別のレベル番号88を割り当てる。

4.3.3.2.2 表の初期値 作業場所節の、表のなかの要素の初期値は、つぎのいずれかによって指定する。

- (1) 表を一つのレコードとし、一連のデータ記述項で記述する。各記述項は、表の要素または要素の部分の値を指定する。レコードとその要素をきめるために、必要な記述句を用いてもよい。この方法は、けたづめやUSAGE（用途）などのために、表の要素を別々に扱いたいときに用いる。表の階層構造は、REDEFINES（再定義）句を含む記述項と、それに従属する記述項を用いて示す。

REDEFINES句を含む記述項のあとの従属する記述項は、OCCURS（反復）句によって反復されるから、そこにVALUE（値）句が書いてあってはならない。

- (2) 表の要素を別々に扱う必要がないときは、表全体の初期値は、表全体をきめる記述項で与えてもよい。低位の記述項で、表の階層構造を示す。低位の記述項にVALUE句が書いてあってはならない。

4.3.3.3 データの字類の概念 データ項目の五つの項類（category, 5.5.7参照）は、英字、数字、英数字の三つの字類（class）にまとめられる。英字と数字については、項類と字類とは同義とする。英数字字類は、英数字編集、数字編集、英数字（編集なし）の三つの項類を含む。基本項目は、どれかの字類およびどれかの項類に属する。集団項目の字類は、それに従属する基本項目の字類が何であっても実行時には英数字として扱われる。つぎに、データ項目の字類と項類との関係を示す。

項目のレベル	字 類	項 類
基 本 項 目	英 字	英 字
	数 字	数 字
	英 数 字	数字編集 英数字編集 英 数 字
集 団 項 目	英 数 字	英 字 数 字 数字編集 英数字編集 英 数 字

備考 字類および項類に属さないデータ項目として指標データ項目がある。

4.3.3.4 文字の表現と基数の選定 数字項目の値は、装置によっては二進法や十進法などの形式で表現してよい。十進法を表現するのに種々の形式をとってよい。これらの表現は、実際にはビットの組合せであるので、二進化十進法とよぶ。基数や形式の選択は、計算機の計算方法による。いくつもの計算基数が使えるときには、その選定は、USAGE(用途)句などによってきめる。二進化十進法は、英数字項目の文字や記号を表わすのにも用いる。

固有の二進化英数字や二進化十進法の選定は、計算機的方式とその外部媒体とによる。

ある計算機でデータを表現するのに、いくつもの形式があるときには、データ記述でとくに指定しないかぎり、標準データ形式が用いられる。外部媒体と計算機の双方が、複数個のデータ表現を扱う能力を有するとき、またはデータに関連する外部媒体がないときは、この選定は、USAGEやPICTUREなどの句による。作成者は、COBOLを作成するにあたって、計算機で用いられる形式を完全に説明しておかなければならない。固有のデータ形式を選定する方法は、プログラマに公表し、利用できるようにしておかなければならない。

基本データ項目や集団項目の大きさは、その項目を標準データ形式で表現したときの文字数である。けたづめを行なった結果、この大きさと内部表現のために必要な実際の文字数とは、異なる場合があってもよい。

4.3.3.5 算術符号 算術符号は、二つの目的で使う。

- (1) 操作される項目の値を示すものを演算符号（operational sign）という。たいていの表現形式では、演算符号を表現するのに標準のやり方がきまっている。したがって、ある項目に演算符号が

ついているかないかだけを指示すれば、ふつうはそれだけで十分である。

- (2) 外部利用のために編集された報告書で、項目の値の正負を指示するものを編集用符号 (editing sign control character) という。編集用符号文字は、項目の符号を表示するのに用い、演算符号ではない。これらの編集用文字は、PICTURE (形式) 句のなかで用いる。

4.3.3.6 実行用プログラムの効率をたかめるための項目のけたづめ 計算機の記憶装置は番地づけのための境界 (語の境界、半語の境界、バイトの境界) をもつように構成されていてよい。データの割付け方法は、実行用プログラムがきめ、番地づけの境界は無視される。

ただし、データのある種の使い方 (算術演算や添字など) では、データが番地づけのための境界にあわせて記憶されているとつごうのよい場合がある。とくに、複数のデータ項目の部分が、となりあう二つの境界のなかに含まれていたり、一つのデータ項目が、境界によって分断されていたりすると、これらのデータを呼び出したり記憶させたりするための余分の機械語命令が実行用プログラム中で必要になる。

このような余分の機械語命令が不要な方法で、データ項目を境界にあわせて配置することを、けたづめという。けたづめされた項目は、その形式にあわせて処理される。けたづめされた形式への変換は、その項目へデータをしまふ手続き (READ や WRITE はのぞく) の実行時にだけおこる。

けたづめは、つぎの二つの方法で行なう。

- (1) SYNCHRONIZED (けたづめ) 句を使う。

- (2) SYNCHRONIZED 句を使わずに、正しい境界を調べてそれにあうようにデータを構成する。

けたづめについて、特殊な方法を用意する作成者は、それらの正しい用法について規定しておかなければならない。

4.3.3.7 一意参照

4.3.3.7.1 修飾 COBOL の原始プログラム中で用いる名前は、同じつづりの名前が他にないか、または名前の系列の上位にあるいくつかの名前をつけくわえて一意にできるかして、すべて一意でなければならない。このとき、上位の名前を修飾語 (qualifier) といい、一意にするこの手順を修飾 (qualification) という。名前を一意にするには、必要な修飾語をつけくわえればよく、上位の名前をすべて書く必要はない。データ名については、レベル指示語 (level indicator) のあとに書いた名前を最高位の修飾語とする。段落名については、節名だけを修飾語として用いることができる。したがって、レベル指示語のあとに書いた名前や節名は、それぞれ一意でなければならない、修飾はできない。添字または指標のついたデータ名や条件変数も、修飾によって一意にすることができる。条件変数の名前を、その条件名の修飾語に使える。修飾の有無にかかわらず、同じつづりの名前をデータ名と手続き名との両方に使ってはならない。

修飾の一般形式は、つぎのとおりとする。IN と OF とは、機能的に同じとする。

書き方 1

$$\left\{ \begin{array}{l} \text{データ名-1} \\ \text{条件名} \end{array} \right\} \left\{ \left\{ \begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right\} \text{データ名-2} \right\} \dots$$

書き方 2

$$\text{段落名} \left\{ \left\{ \begin{array}{l} \text{OF} \\ \text{IN} \end{array} \right\} \text{節名} \right\}$$

修飾に関する規則は、つぎのとおりとする。

- (1) 各修飾語は、それが修飾する名前と同じ階層系列に属し、低いレベルのものから順に書く。
(2) 同じ系列中の二つのレベルに、同じ名前を書いてはならない。

- (3) プログラムを書くときに、複数のデータ項目に同じデータ名や条件名を割り当てた場合は、そのデータ名や条件名を手続き部、環境部、データ部で参照するには必ず修飾する。ただし、**REDEFINES** (再定義) 句では、修飾を必要とするような書き方はできない。
- (4) 一つの節のなかに同じ段落名を二つ以上書けない。段落名を節名で修飾するとき、**SECTION** という語は書かない。ある節のなかで、そのなかの段落名を参照するときは、修飾しなくてもよい。
- (5) データ名を修飾語として使うときは、添字はつけられない。
- (6) 修飾する必要のない名前を修飾してもよい。一意にする修飾語の組合せがいくつもあるとき、どれを使ってもよい。

4.3.3.7.2 添字づけ 個別的なデータ名をもたない同種の要素を列挙した表のなかの、個々の要素を参照するのに添字 (subscript) を用いる。添字は、整数値、特殊レジスタまたはデータ名で表わす。このデータ名は、整数を表わす基本項目とする。添字をデータ名で表わすときに、このデータ名は修飾してあってもよいが、添字をつけてあってはならない。

添字は、正号をもっているもよい。添字のとりうる値の最小値は、1 とする。この値は、表の最初の要素をさす。添字の値 2, 3, ... は、表のそのつぎ以降の要素を順々にさす。添字のとりうる値の最大値は、**OCCURS** (反復) 句で指定された項目の反復回数の最大値とする。

添字または添字の組は、表の要素のデータ名のあとに空白をあけ、そのあとにかっこで囲んでつける。表の要素のデータ名に添字をつけたものを、添字つきデータ名または一意名という。かっこのなかに複数の添字を書くときは、各添字の間はコンマと空白で区切る。左かっこと左端の添字、右かっこと右端の添字の間は、それぞれ空白をあけてはならない。

添字づけの一般形式は、つぎによる。

データ名 (添字 [, 添字] ...)

参考 COBOL 1969 では、添字づけおよび指標づけのコンマは書いても書いてもよい。かっこの側に空白をあけてもよい。

4.3.3.7.3 指標づけ 指標づけ (indexing) を指定して、同種の要素からなる表のなかの個々の要素を参照することができる。表を参照するとき、**INDEXED BY** (指標つき) 句を用いて表の各レベルに指標 (index) を割り当てる。**INDEXED BY** 句で与えられた名前を指標名 (index-name) といい、割り当てられた指標を参照するのに用いる。指標名を使って表を参照するためには、**SET** (設定) 命令などで初期値を与えておかなければならない。

指標名を添字と同じ形式で使う方法を、直接指標づけ (direct indexing) という。指標名のあとに、減算記号または一と符号のつかない整数を書き、その全体をかっこで囲んで、データ名のうしろに空白をあけて続けた形式で使う方法を、相対指標づけ (relative indexing) という。

指標づけの一般形式は、つぎによる。

データ名 (指標名 $\left[\left\{ \begin{array}{c} + \\ - \end{array} \right\} \text{整数} \right] \left[, \text{指標名} \left\{ \begin{array}{c} + \\ - \end{array} \right\} \text{整数} \right] \dots$)

参考 COBOL 1969 では、指標のうちのいくつかが単に整数値であってもよい。

4.3.3.7.4 一意名 一意名 (identifier) とは、データ名がプログラム中で一意でないときに、いくつかの修飾語、添字、指標を構文的に正しくつけて一意にしなければならないことを示す用語である。

一意名の一般形式は、つぎによる。

書き方 1

データ名-1 $\left[\left\{ \begin{array}{c} \text{OF} \\ \text{IN} \end{array} \right\} \right.$ データ名-2 $\left. \right] \cdots$
 [(添字-1 [(, 添字-2 [(, 添字-3))])]]

書き方 2

データ名-1 $\left[\left\{ \begin{array}{c} \text{OF} \\ \text{IN} \end{array} \right\} \right.$ データ名-2 $\left. \right] \cdots$
 $\left[\begin{array}{l} \left(\text{指標名-1} \left[\left\{ \begin{array}{c} + \\ - \end{array} \right\} \right. \right. \text{整数} \left. \left. \right] \right. \\ , \text{指標名-2} \left[\left\{ \begin{array}{c} + \\ - \end{array} \right\} \right. \text{整数} \left. \left. \right] \right. \\ , \text{指標名-3} \left[\left\{ \begin{array}{c} + \\ - \end{array} \right\} \right. \text{整数} \left. \left. \right] \right] \right] \right]$

修飾, 添字づけ, 指標づけの制限は, つぎのとおりとする。

- (1) 一般形式中のコンマは, 必ず書かなければならない。
- (2) データ名が, 指標, 添字, 修飾語として用いられているときは, そのデータ名自身には添字も指標もつけられない。
- (3) 添字づけのできない場合には, 指標づけも書いてはならない。
- (4) 指標は, SET (設定), SEARCH (表引き), PERFORM (実行) 命令でしか変更できない。USAGE IS INDEX (用途は指標) 句で記述されたデータ項目には, 指標名の値が作成者のきめた形式で変換なしにたかわえられる。このようなデータ項目を指標データ項目という。

4.4 見出し部

4.4.1 概説 COBOL の原始プログラムには, それぞれ見出し部 (identification division) を書かなければならない。これは, 原始プログラムや翻訳出力印刷の見出しとなるものとする。利用者は, 以下に示す一般形式の段落 (paragraph) のなかに, プログラムを書いた日付け, プログラムを翻訳した日付けなどを書く。

4.4.2 構成 きまった段落名 (paragraph-name) を用いて, その段落に含まれている情報の見出しとする。最初の段落としてプログラム名 (program-id) 段落をおき, ここにプログラムの名前を必ず書かなければならない。これ以外の段落は任意で, 以下の一般形式で示す順序にしたがってこの部を含めることができる。

4.4.3 構造 見出し部の書き方は, つぎの一般形式による。

IDENTIFICATION DIVISION.

PROGRAM-ID. プログラム名.

AUTHOR. [注記項] $\cdots$

INSTALLATION. [注記項] $\cdots$

DATE-WRITTEN. [注記項] $\cdots$

DATE-COMPILED. [注記項] $\cdots$

SECURITY. [注記項] $\cdots$

REMARKS. [注記項] $\cdots$

参考 COBOL 1969 には, REMARKS 段落はない。

4.5 環境部

4.5.1 概説 環境部 (environment division) は、データ処理の問題をCOBOL で記述するときに、使用する計算機の機械的な特性によってきまる性質を書き表わす標準的な方法を定める。この部では、翻訳用計算機および実行用計算機の構成を指定する。そのほか、入出力管理、特殊なハードウェアの特性や制御技法に関する情報を与えることができる。

環境部は、COBOL の原始プログラムに必ず書かなければならない。

4.5.2 構成 環境部には、構成節 (configuration section) と入出力節 (input-output section) との二つの節を書く。

構成節には、翻訳用計算機と実行用計算機との特性を書く。この節は、つぎの三つの段落からなる。翻訳用計算機 (source-computer) 段落では、原始プログラムを翻訳する計算機の構成を指定し、実行用計算機 (object-computer) 段落では、翻訳されたプログラムを実行する計算機の構成を指定する。特殊名 (special-names) 段落では、原始プログラムで使う呼び名 (mnemonic-name) と翻訳ルーチンが扱う作成者のきめた名前 (implementor-name) との関係指定する。

入出力節では、外部媒体と実行用プログラムの間で、データを転送し処理するのに必要な情報を書く。この節は、つぎの二つの段落からなる。ファイル管理 (file-control) 段落では、ファイルに名前をつけ、外部媒体を割り当てる。入出力管理 (i-o-control) 段落では、実行用プログラムで使われる特殊な制御技法を指定する。

4.5.3 構造 環境部の節と段落との書き方は、つぎの一般形式による。原始プログラムには、ここに示した順序で書かなければならない。

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. 翻訳用計算機記述項

OBJECT-COMPUTER. 実行用計算機記述項

(SPECIAL-NAMES. 特殊名記述項)

(INPUT-OUTPUT SECTION.

FILE-CONTROL. { ファイル管理記述項 })

(I-O-CONTROL. 入出力管理記述項))

4.6 データ部

4.6.1 概説 データ部 (data division) では、実行用プログラムが入力として受け取り、操作し、作り出し、あるいは出力として書き出すデータについて記述する。処理されるデータは、二つに分類する。

- (1) ファイルに記録されるもので、指定された領域から計算機の内部記憶に読み込まれたり、そこから書き出されたりするもの。
- (2) 内部で展開され、中間場所すなわち作業場所におかれるデータ、報告書作成の目的できまった形式に編集されるデータ、および利用者が指定した定数。

4.6.2 データ記述の物理的論理的構成

4.6.2.1 データ部の構成 データ部は、プログラムに必ず書かなければならない部の一つで、節に細分される。節 (section) は、ファイル節、作業場所節および報告書節とする。

ファイル節 (file section) では、外部媒体に記録されるデータファイルの内容を定義する。各ファイルは、一つのファイル記述と、それに続く一つ以上のレコード記述によって定義する。報告書出力ファイルとして使

うファイルのときには、ファイル記述だけで、レコード記述は書かない。作業場所節 (working-storage section) では、外部データファイルに属さず、内部的に処理され展開されるデータ項目、および値が原始プログラムで与えられ、実行用プログラムの実行中に変わることのないデータ項目について記述する。作業場所節でも、レコードと、レコードにまとめない独立項目との両方を指定することができる。報告書節 (report section) では、作成される報告書の内容や形式を記述する。

4.6.2.2 データ部の構造 データ部は、正書法にしたがって書く。データ部は、つぎの見出しで書き始めなければならない。

DATA DIVISION.

データ部の各節で、不必要なものは原始プログラムに書く必要はない。節のきまった見出しと、データ部に書くきまった順序とはつぎのとおりとする。

FILE SECTION.

WORKING-STORAGE SECTION.

REPORT SECTION.

ファイル節と報告書節とは、それぞれの節の見出しのあとに、ファイル句からなるいくつかの記述項と、それに関連あるレコード記述項 (record description entry) とを書いて構成する。作業場所節は、節の見出しのあとに、独立項目 (noncontiguous item) のデータ記述項 (data description entry) をおき、そのあとにレコード記述を書いて構成する。

4.7 手続き部

4.7.1 概説 COBOL の原始プログラムには、手続き部 (procedure division) がなければならない。手続き部は、宣言部分と手続き部分からなる。

4.7.1.1 宣言部分 手続き部の始めに必要語 **DECLARATIVES** を書き、宣言部分 (declaratives) の節 (section) を続けて書いて、必要語 **END DECLARATIVES** を書いてとめる。

USE (使用) 命令を宣言命令という。

4.7.1.2 手続き部分 手続き部における手続き部分 (procedure) は、一つの段落、連続したいくつかの段落の組、一つの節、または連続したいくつかの節の組で構成する。ある段落がどれかの節に属するときは、すべての段落は、節にまとめて書く。手続き名とは、原始プログラム中に書いた段落や節を参照するときに使う語である。手続き名は、段落名、修飾された段落名、または節名とする。

手続き部の終り、すなわちプログラムの物理的な終りとは、COBOL の原始プログラムの書き終りで、それ以降、手続きを書かない物理的な位置とする。

節 (section) は、節の見出しとそのあとの連続したいくつかの段落からなる。節の終りは、つぎの節名の直前、手続き部の終り、または手続き部の宣言部分では、必要語 **END DECLARATIVES** とする。

段落 (paragraph) は、段落名とそのあとの連続したいくつかの文からなる。段落の終りは、つぎの段落名の直前、手続き部の終り、または手続き部の宣言部分では、必要語 **END DECLARATIVES** とする。

文 (sentence) は、いくつかの命令からなり、終止符と空白で終わる。

命令 (statement) とは、COBOL の動詞のあとに、語と記号とを構文的に正しく組み合わせて書いたものとする。

一意名 (identifier) とは、データ項目を一意に参照するのに必要な語または語の組とする。

4.7.1.3 実行 実行は、手続き部の宣言部分でない最初の命令から始まる。命令群は、原則として原始プログラムに書いてある順序で実行される。ただし、その順序と異なる場合は規則で示す。

4.7.1.4 手続き部の構造

4.7.1.4.1 手続き部の見出し 手続き部の最初には、つぎの見出しを書かなければならない。

PROCEDURE DIVISION.

4.7.1.4.2 手続き部の本文の構造 手続き部の本文は、つぎのいずれかの形式で書かなければならない。

書き方 1

[DECLARATIVES.

{ 節名 SECTION. 宣言文

{ 段落名. { 文 } . . . } . . .

END DECLARATIVES.)

{ 節名 SECTION (優先番号).

{ 段落名. { 文 } . . . } . . .

書き方 2

{ 段落名. { 文 } . . . } . . .

4.7.2 命令と文 命令は、無条件命令、条件命令、翻訳指示命令の3種類とする。

文は、無条件文、条件文、翻訳指示文の3種類とする。

4.7.2.1 条件命令と条件文

4.7.2.1.1 条件命令の定義 条件命令 (conditional statement) は、条件の真理値を定め、実行用プログラムのつぎにとる動作がこの真理値に左右されることを指定する。

条件命令は、IF, READ, SEARCH, RETURN命令, INVALID KEY 指定のあるWRITE 命令, SIZE ERROR 指定のある算術命令 (ADD, COMPUTE, DIVIDE, MULTIPLY, SUBTRACT) とする。

4.7.2.1.2 条件文の定義 終止符と空白でとめられた条件命令を、条件文という。条件命令の前に無条件命令があってもよい。

4.7.2.2 翻訳指示命令と翻訳指示文

4.7.2.2.1 翻訳指示命令の定義 翻訳指示命令 (compiler-directing statement) は、翻訳指示動詞とその作用対象 (operand) とからなる。翻訳指示動詞は、COPY, ENTER, NOTE, USE とする。

4.7.2.2.2 翻訳指示文の定義 終止符と空白でとめられた翻訳指示命令を、翻訳指示文という。

4.7.2.3 無条件命令と無条件文

4.7.2.3.1 無条件命令の定義 無条件命令 (imperative statement) は、実行用プログラムのとる特定の動作を指示する。

無条件命令は、条件命令、翻訳指示命令以外のすべての命令とする。

無条件命令は、いくつかの無条件命令のつながったものでもよく、それらの間を分離符で区切ってもよい。無条件動詞は、つぎのとおりとする。

ACCEPT	DISPLAY	GO	PERFORM
ADD ⁽¹⁾	DIVIDE ⁽¹⁾	INITIATE	RELEASE
ALTER	EXAMINE	MOVE	SEEK
CLOSE	EXIT	MULTIPLY ⁽¹⁾	SET
COMPUTE ⁽¹⁾	GENERATE	OPEN	SORT

STOP

SUBTRACT⁽¹⁾

TERMINATE

WRITE⁽²⁾

注 (1) SIZE ERROR 指定のないもの。

(2) INVALID KEY 指定のないもの。

命令の一般形式中で、無条件命令と指定してある場合、その無条件命令とは、終止符、IF 動詞が先行している場合の ELSE、または SEARCH 動詞が先行している場合の WHEN で終わるいくつかの連続した無条件命令のこととする。

4.7.2.3.2 無条件文の定義 無条件命令を終止符と空白でとめたものを無条件文という。

4.8 正書法

4.8.1 概説 正書法 (reference format) は、COBOL の原始プログラムを記述するための標準の方法であって、入出力媒体上の行を構成する文字位置を基準として定める。作成者は、その翻訳ルーチンが使用する入出力媒体について、行や文字位置の意味を定義しておかなければならない。それらの定義にしたがって翻訳ルーチンは、正書法で書かれた原始プログラムを受け入れ、正書法で出力印刷を作り出す。

正書法に関する空白あけの規則は、他のすべての空白あけの規則に優先する。

原始プログラムの各部は、見出し部、環境部、データ部、手続き部の順に書かなければならない。各部は正書法の規則にしたがって書く。

4.8.2 正書法の表現 1 行の正書法は、つぎのように表わす。

欄 L						欄 C		欄 A		欄 B				欄 R
1	2	3	4	5	6	7	8	9	10	11	12	13	.	.

欄 L は、1 行の最左端の文字位置を示す。

欄 C は、L から数えて 7 文字目の位置を示す。

欄 A は、L から数えて 8 文字目の位置を示す。

欄 B は、L から数えて 12 文字目の位置を示す。

欄 R は、1 行の最右端の文字位置を示す。

一連番号領域は、欄 L から始まる 6 文字とする。

つなぎ領域は、欄 C の 1 文字分とする。

A 領域は、欄 A から始まる 4 文字分とする。

B 領域は、欄 B から始まる作成者のきめた有限個の文字分とする。

4.8.2.1 一連番号 一連番号 (sequence number) は、一連番号領域に書かれた 6 けたの数字からなり、原始プログラムの行の識別に使う。

4.8.2.2 行のつなぎ 文や記述項が 2 行以上になるときは、その続き部分は B 領域に書く。この続きの行をあとの行 (continuation line) とよぶ。続けられる行を前の行 (continued line) とよぶ。任意の語や直定数を、あとの行に分けて書いてもよい。

ある行のつなぎ領域にハイフン (—) を書くとき現在の行の B 領域の空白でない最初の文字が、前の行の空白でない最後の文字のあとに、間の空白には関係なく続くことを示す。ただし、前の行にまだ引用符で閉じてない文字直定数がある場合には、あとの行の B 領域の空白でない最初の文字は、引用符でなければならぬ。そして、文字直定数の続きは、この引用符のすぐつぎの文字位置から書く。前の行の終りまでの空白はすべてその直定数の部分であるとなされる。あとの行の A 領域は空白でなければならない。

つなぎ領域にハイフンがないと、前の行の最後の文字のあとに空白があるものとみなされる。

4.8.2.3 空白行 欄Cから欄Rまでがすべて空白の行を、空白行 (blank line) という。空白行は、あとの行の直前以外ならば、原始プログラム中のどこにあってもよい。

参考 COBOL 1969 では、欄Cに星印または斜線を記入した行は、注記行とされる。

4.8.3 部、節、段落の正書法

4.8.3.1 部の見出し 部の見出しは、部名をA領域から書き始め、つぎに空白、語 **DIVISION**、終止符を書く。部の見出しのあとには、続く節の見出し、段落の見出しまたは段落名がくるまで（すなわち、つなぎ領域にハイフンがなく、A領域に空白以外の文字がある行の前まで）は何も書いてはならない。

4.8.3.2 節の見出し 節の名前は、A領域から書き始め、つぎに空白、語 **SECTION** を書き、必要ならば空白に続けて優先番号を書き、終止符と空白でとめる。節の見出しのあとには、続く段落の見出しや段落名がくるまで（すなわち、つなぎ領域にはハイフンがなく、A領域に空白以外の文字がある行の前まで）は何も書いてはならない。ただし、つぎの文は書いてもよい。

(1) 手続き部の宣言部分では、節の見出しのあとに **USE** (使用) 文を書いてよい。

(2) **COPY** (複写) 命令を書いてよい。

節は、環境部と手続き部とでは段落の集まりからなり、データ部では、データ部の記述項の集まりからなる。見出し部には、節名なしに段落名を書く。

4.8.3.3 段落の見出し、段落名および段落 段落は、段落名のあとに一つ以上の文を書いたものか、または段落の見出しのあとに一つ以上の記述項を書いたものからなる。

段落の見出しは、部や節の最初の行のあとに任意の行のA領域から書き始め、終止符と空白でとめる。

段落の最初の文や記述項は、段落名や段落の見出しを書いた行か、またそのあとの空白行でない行のB領域から書き始める。続く文や記述項は、前の文や記述項と同じ行か、またはそのあとの空白行でない行のB領域から書き始める。

文は、一つ以上の命令からなる。記述項は、一つ以上の句からなる。すべての文や記述項は、終止符と空白でとめる。

段落中の文や記述項を2行以上に書く必要があるときは、4.8.2.2による。

4.8.4 データ部の記述項 データ部の各記述項は、レベル指示語またはレベル番号とそれに続く空白で始まり、それにデータ項目の名前が続き⁽¹⁾、さらにそのデータ項目について記述する一連の独立な句が続く。各句は、この記述項の最後の句を除いて、セミコロンと空白で区切ってもよい。最後の句は、常に終止符と空白で終わる。

注⁽¹⁾ 報告書節では、データ項目の名前は省略することができる。

データ部の記述項には、レベル指示語で始まるものと、レベル番号で始まるものがある。

レベル指示語 (level indicator) は、**FD**, **SD**, **RD**とする。

レベル指示語で始まるデータ部の記述項では、レベル指示語は、A領域から書き始め、それに続いてデータ名と適当な記述情報とをB領域に書く。

レベル番号 (level-number) で始まるデータ部の記述項を、データ記述項 (data description entry) という。

レベル番号01または77で始まるデータ記述項では、レベル番号をA領域から書き始め、続いてそのレコード名または項目名と適当な記述情報とをB領域に書く。

レベル番号は01から49まで、または66, 77もしくは88のいずれかとする。一けたのレベル番号は、一けたの空白のあとにその数字を書いてもよいし、ゼロのあとにその数字を書いてもよい。レベル番号とそのつぎの

語との間には、少なくとも1個の空白をおかなければならない。

引き続きデータ記述項は、最初のものと同じ形式にしてもよいし、レベル番号にしたがって階段状に書いてもよい。各記述項の出力印刷を階段状にするのは、入力が階段状になっているときだけである。階段状にしても、レベル番号の大きさには影響しない。

レベル番号を階段状にすると、新しいレベル番号は、A領域の右側に任意の個数の空白をあけた位置から書き始めてよい。階段状の出し入れの深さは、物理的媒体の幅によって制限されるだけである。

引き続き、より大きい値をもつレベル番号をすべて階段状にする必要はなく、前のレベル番号と同じ文字位置から始めてもよい。

4.8.5 宣言部分 手続き部の宣言部分の始めと終りを示す予約語 **DECLARATIVES** や **END** **DECLARATIVES** を書いた行には、それ以外に何も書いてはならない。それぞれは、A領域から書き始め、語 **DECLARATIVES** のあとで終止符と空白でとめる。

4.9 予約語 予約語 (reserved word) は、つぎのとおりとする。

ACCEPT	CLOCK-UNITS	DIVIDE	GREATER
ACCESS	CLOSE	DIVISION	GROUP
ACTUAL	COBOL	DOWN	HEADING
ADD	CODE	ELSE	HIGH-VALUE
ADDRESS	COLUMN	END	HIGH-VALUES
ADVANCING	COMMA	ENDING	I-O
AFTER	COMP	ENTER	I-O-CONTROL
ALL	COMPUTATIONAL	ENVIRONMENT	IDENTIFICATION
ALPHABETIC	COMPUTE	EQUAL	IF
ALTER	CONFIGURATION	ERROR	IN
ALTERNATE	CONTAINS	EVERY	INDEX
AND	CONTROL	EXAMINE	INDEXED
ARE	CONTROLS	EXIT	INDICATE
AREA	COPY	FD	INITIATE
AREAS	CORR	FILE	INPUT
ASCENDING	CORRESPONDING	FILE-CONTROL	INPUT-OUTPUT
ASSIGN	CURRENCY	FILE-LIMIT	INSTALLATION
AT	DATA	FILE-LIMITS	INTO
AUTHOR	DATE-COMPILED	FILLER	INVALID
BEFORE	DATE-WRITTEN	FINAL	IS
BEGINNING	DE	FIRST	JUST
BLANK	DECIMAL-POINT	FOOTING	JUSTIFIED
BLOCK	DECLARATIVES	FOR	KEY
BY	DEPENDING	FROM	
CF	DESCENDING	GENERATE	LABEL
CH	DETAIL	GIVING	LAST
CHARACTERS	DISPLAY	GO	LEADING

LEFT	OUTPUT	RESERVE	SUBTRACT
LESS	PAGE	RESET	SUM
LIMIT	PAGE-COUNTER	RETURN	SYNC
LIMITS	PERFORM	REVERSED	SYNCHRONIZED
LINE	PF	REWIND	TALLY
LINE-COUNTER	PH	RF	TALLYING
LINES	PIC	RH	TAPE
LOCK	PICTURE	RIGHT	TERMINATE
LOW-VALUE	PLUS	ROUNDED	THAN
LOW-VALUES	POSITION	RUN	THROUGH } 同義
MEMORY	POSITIVE	SAME	THRU
MODE	PROCEDURE	SD	TIMES
MODULES	PROCEED	SEARCH	TO
MOVE	PROCESSING	SECTION	TYPE
MULTIPLE	PROGRAM-ID	SECURITY	UNIT
MULTIPLY	QUOTE	SEEK	UNTIL
NEGATIVE	QUOTES	SEGMENT-LIMIT	UP
NEXT	RANDOM	SELECT	UPON
NO	RD	SENTENCE	USAGE
NOT	READ	SEQUENTIAL	USE
NOTE	RECORD	SET	USING
NUMBER	RECORDS	SIGN	VALUE
NUMERIC	REDEFINES	SIZE	VALUES
OBJECT-COMPUTER	REEL	SORT	VARYING
OCCURS	RELEASE	SOURCE	WHEN
OF	REMARKS	SOURCE-COMPUTER	WITH
OFF	REMAINDER	SPACE	WORDS
OMITTED	RENAMES	SPACES	WORKING-STORAGE
ON	REPLACING	SPECIAL-NAMES	WRITE
OPEN	REPORT	STANDARD	ZERO
OPTIONAL	REPORTING	STATUS	ZEROES
OR	REPORTS	STOP	ZEROS
	RERUN		

付録 3 情報処理用語

【I D P (集^{あいでーびー}データ処^{しゅうちゆう}理^{しより}) integrated data processing】

種々のデータを1か所に集め、ある目的に従ってそれぞれに必要な処理(計算)をし、必要ならば結果を返送すること。

【I S O International Organization for Standardization】

国際標準化機構。商品やサービス交換を容易にするために、国際的な標準化を促進している機関。

【アクセス access】

記憶装置と入出力装置との間で、データを転送すること。

【アクセス・アーム access arm】

磁気ディスクや磁気ドラムの装置の一部分で、データを読み書きするための機構。読取りヘッドと書込みヘッドを持っている。

【アクセス時間^{じかん} access time】

制御装置が記憶装置との間の情報の転送を要求してから、転送が実際に開始されるまでの時間。

たとえば、 n 語の情報をたぐわえることができる磁気ドラムのトラックでは、条件が一番良いときはすぐに転送できるから呼出し時間はゼロであり、最悪の場合は転送するまでに $(n-1)$ 語時間かかる。したがって、平均呼出し時間は $\frac{1}{2}(n-1)$ 語時間になる。

【アセンブラ assembler】

記号言語(アセンブラ言語)でコーディングされたプログラムを機械語に変換する言語翻訳プログラム。

【アSEMBル assemble】

記号言語でコーディングされたプログラムを翻訳して、機械語プログラムを作成すること。

【アドレス(番地)^{ばんち} address】

転送する情報の出所や行先を表わす表示。通常、記憶装置のなかの特定の場所を指定するのに用いる。

【アドレス部^ぶ address part】

命令語のなかでアドレスを指定している部分。機械語ではオペランドがアドレス部になる。

【アドレス変更(修飾)^{へんこう しゅうしよく} address modification】

命令語のアドレス部について指標(インデックス)レジスタを操作したり、演算を施したりして、内容を変更すること。

【アドレス・レジスタ address register】

アドレスが記憶されるレジスタ。

【アナログ型コンピュータ ^{がた} analog computer】

主としてアナログ表現でデータを取扱うコンピュータ。

【粗い流れ図 ^{あらながず} general flowchart】

仕事の流れや、処理手順を大まかに書表わした流れ図。

【ALGOL ^{あるどる} Algorithmic Language】

国際的に用いられている技術計算用の問題向き言語。

【RPG ^{あーるびーじー} report program generator】

報告書作成用として広く使われているプログラム言語。

【アンパック unpack】

パックされたデータを元の形のデータに戻すこと。

【一括処理 ^{いつかつしより} (バッチ処理 ^{しより}) batch processing】

データ処理の場合に、処理するデータをためておき、ひとまとめにしてから処理する方式。

【EDP ^{いーでーびー} electronic data processing】

コンピュータによる自動データ処理。

【EDPS ^{いーでーびーえす} electronic data processing system】

コンピュータと同義語。

【移動 ^{いどう} (する) move】

ある場所からデータを送り、他の場所でそれを受取ること。

【入口点 ^{いりぐちてん} entry point】

あるルーチンから他のルーチンに制御を渡すとき、受取り側のルーチンの中で最初に制御を受ける点。

【印刷装置 ^{いんさつそうち} printer】

コンピュータの出力装置の1つ。コンピュータから情報を表わす信号を受けて、それに対応する文字を出力用紙に記録する。

【インタプリタ ^{つうやく} (通訳ルーチン) interpreter】

原始言語の表現を、1つずつ順番に翻訳しながら実行するプログラム。

【英数字コード ^{えいすうじ} alphanumeric code】

英字、数字、特殊文字から成る文字セット。

【映像表示装置 ^{えいぞうひょうじそうち} graphic display unit】

えくす　　こう

通常の紙カードの第2段目のせん孔。

次表参照

【エラー・メッセージ error message】

おとうよう

特定の分野の業務を処理するために開発された汎用プログラム。LP, PERT, GPSSな

とはその例である。

すでにせん孔してあるけたに、さらに重複してせん孔すること。

せん孔がカードの正確な位置からずれること。

(1) 情報の転送過程で、人手の介入を必要とする状態。

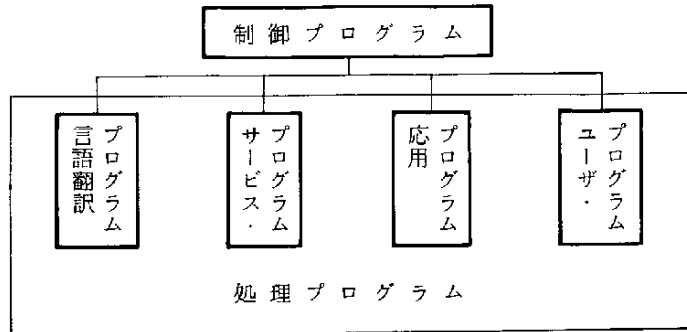
(2) 中央演算処理装置の直接制御下でない状態。

(1) 機械語や記号言語のアドレス部。

(2) 問題向き言語では、演算される数。たとえば、 $A+B$ という演算では、 A は第 1 オペランド、 B は第 2 オペランドと呼ばれる。

【オペレーティング・システム operating system】

コンピュータを操作するために必要な技法や処理手順を、組織的に集約したもの。
一般的には次のように体系立てられたプログラムの集合である。



おんせいおうとう

【音声応答システム audio response system】

磁気ドラム装置や磁気ディスク装置に記録された情報を用いて、電話などによる照会に音声で
応答する装置。

【オンライン on-line】

- (1) 情報を転送する過程で、人手の介入を必要としない状態。
- (2) 中央演算処理装置の制御下にある状態。

【オンライン・システム on-line system】

入力データを、通信回線を介して、人手の介入なしに発生地点からコンピュータへ送ったり、
出力データを、それを必要としている場所に直接伝送したりするシステム。

【オンライン・バンキング・システム on-line banking system】

オンライン・リアルタイム・システムの1つで、中央のコンピュータに接続された通信回線で、
銀行の本店と支店の窓口を直結させ、窓口業務の集中処理を行なうシステム。

かいけいき

【会計機 accounting machine】

- (1) 会計記録を作成するけん盤作動の機械。
- (2) 紙カードまたは紙テープなどの入力媒体からデータを読み取り、会計記録や集計表を、連続
した用紙に自動的に印刷する機械。

かい たんい

【下位けた(単位けた) low-order position】

1単位の数字や語の一番右側のけた。

かいてん ま じかん

【回転待ち時間 rotational delay time】

磁気ディスクや磁気ドラムにおいて、アクセス・アームをセットしてから読取りを始めるまでの時間。

【解読器(デコーダ) かいどくき decoder】

複数の入力端子と複数の出力端子とを持つ装置で、入力端子のある組合せに信号が加えられたとき、その組合せに対応する1つの出力端子に信号が現れるもの。符号器と逆の作用をする。

【外部記憶 がいぶきおく external storage】

中央演算処理装置とは切離されているが、中央演算処理装置が受取れる形で情報を保持できる媒体。

たとえば、紙テープ、紙カード、磁気テープ、磁気ディスク、磁気ドラムなどがある。

【カウンタ(計数器) けいすうき counter】

(1) レジスタの一種で、入力信号を受けることによって内容が1ずつ増加したり、減少したりするように構成されたもの。

(2) プログラムの中で、数を数えたり、数の合計を計算したりするために確保する領域。

【拡張2進化10進コード かくちようにしんかじつしん EBCDIC えぶしでいつく】

対応する英語は、extended binary coded decimal interchange codeの略。

文字コードの一種で、現在もっとも広く使われている。

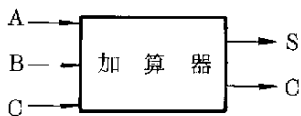
【加算器 かさんき adder】

(1) 2つの数の和を作る回路。

(2) 3個の入力端子と2個の出力端子とを持ち、出力信号が入力信号に対し、次の表の関係にある回路。

表

入 力			出 力	
A	B	C*	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



この回路は2進加算器の1けた分を構成し、各A、Bは演算数の1けた、 C^* は下位からのけた上げ、Sは和のけた、Cは上位へのけた上げに相当する。

【仮数 mantissa】

数字を浮動小数点数で表示するとき、10の x 乗が掛けられる固定小数点係数のこと。

たとえば、 0.125×10^5 で、0.125が仮数。

【カード・イメージ card image】

せん孔カードの内容と1対1で対応した表記形式。

【カード送り card feed】

紙カードを順番に機械の中に送り込む機構。

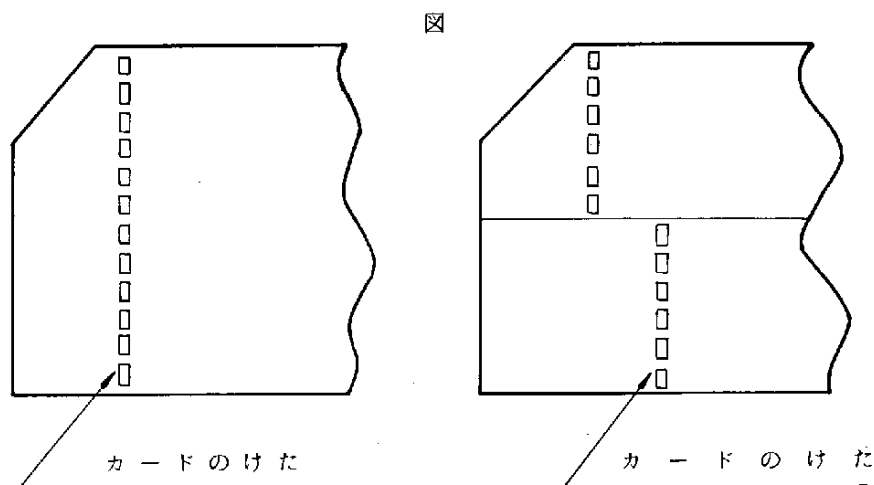
【カード・ゲージ card gauge】

80欄カード全部のせん孔位置が正確に刻まれている金属板で、せん孔されたカードのせん孔位置が正確かどうかをチェックするために使われる。

【カードのけた(欄) card column】

紙カード上で通常1字を記録するために割当てられた一群のせん孔位置。

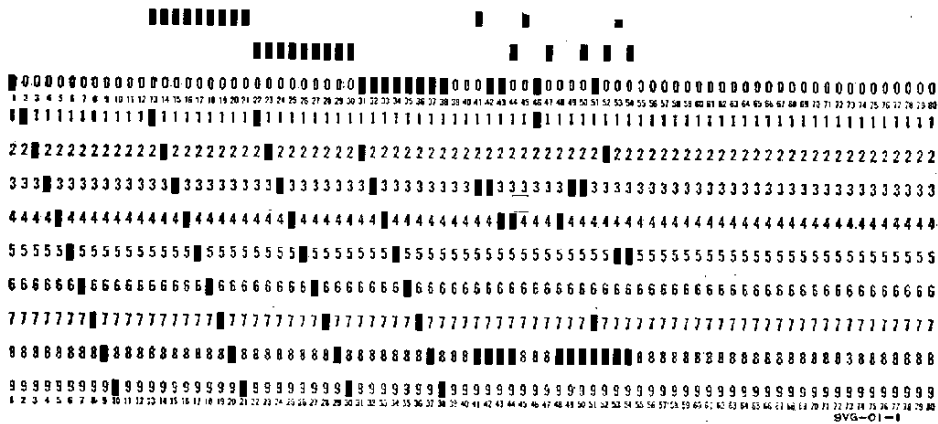
80欄の紙カードでは1けたに12個、90欄の紙カードでは6個の縦に並ぶせん孔位置がある。



【カード・コード card code】

紙カードに文字や数字をせん孔するときにデータを表現するために用いられるコード。

一般的には次のようなせん孔の組合せで文字や数字を表現している。



【カード・システム card system】

コンピュータの外部記憶として紙カードだけが使えるシステム。

【カード・ジャム card jam】

機械の中に紙カードがつかえること。

【カード・スタッカ card stacker】

カード読取せん孔装置で読込んだり、せん孔した紙カードを積重ねて行く機構。

【カード・デッキ card deck】

せん孔カードの集り。

【カード・ファイル card file】

紙カードにせん孔して記録してあるファイル。

【カード・ホッパ card hopper】

カード読取せん孔装置で、処理前の紙カードをカード送り機構に送り込むために積上げておく装置。

【カード読取せん孔装置 card reader punch】

カード読取機構とカードせん孔機構を持つ装置。

【カード読取装置(カード・リーダ) card reader】

カードに記録されている情報を読取る装置。

【可変長 variable length】

ファイルを構成しているレコードの長さが一定でないこと。

【可変長レコード variable length record】

ファイルを構成しているレコードの長さが一定でないときのレコードに対する呼称。

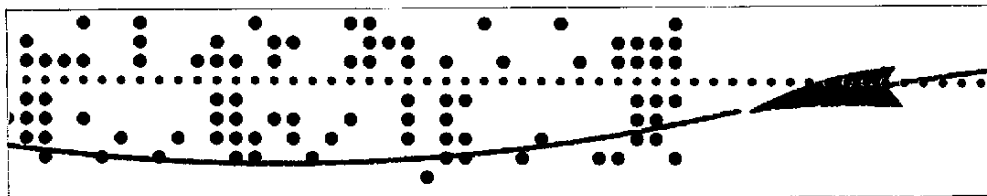
【紙カード (カード) paper card】

一定の形状、寸法の紙のカードで、これに一定の規則に従ってせん孔を行ない、文字、数字、記号を記録し、コンピュータによる情報処理の媒体として用いる。せん孔する代りに、マークを付けて情報を記録することもある。

【紙テープ paper tape】

データを記録するための媒体の1つで、紙カードとともに広く使われている。1けたのせん孔位置の数によって、5単位、6単位、7単位、8単位の4種類のテープがある。

8単位テープの例



【紙テープせん孔装置 paper tape punch】

紙テープにデータをせん孔する装置。

【紙テープ読取装置 paper tape reader】

せん孔された紙テープを読取る装置。

【緩衝域 (バッファ) buffer】

内部記憶と補助記憶や外部記憶の間、入出力装置と内部記憶の間でデータを転送するときに、一時的にあるいは中間的に必要な記憶域や記憶装置。

【監視プログラム supervisor】

システム資源や処理プログラムのスケジュールを決めたり、割付けをしたり、制御をしたりする働きを持っているプログラムに付けられた一般的な名称。

【関数 function】

(1) ある変数の値にもとづいて他の変数の値を定義する関係のこと。たとえば $Y = 3X^2 + 2X$

(2) FORTRANでは特別なサブプログラムに付けられた名称。

【記憶 storage】

必要な情報を主記憶や補助記憶にたくわえること。記憶装置は必要に応じて希望の情報を速や

かにたくわえたり（書込み）、たくわえられた情報を速やかに取出したり（読出し）、消したりできる特性を持つことが必要である。

き おく そう ち
【記憶装置 memory, storage unit】

コンピュータの構成要素の1つで、計算処理に必要な情報を記憶する装置。磁心記憶装置、磁気フィルム記憶装置、磁気ディスク記憶装置、磁気ドラム記憶装置、磁気テープ装置などが広く用いられている。

き お く
【記憶ダンプ storage dump】

内部記憶や補助記憶から外部記憶へ、記憶内容の全部あるいは一部分を複写すること。

き お く よ う り よ う
【記憶容量 storage capacity】

記憶装置にたくわえることが可能な情報量。通常、語数、字数、バイト数、ビット数などで表わす。

き お く
【記憶レジスタ storage register】

記憶装置からのデータや、記憶装置へ送るデータを、一時的にたくわえておくレジスタ。

き か い ご
【機械語 machine language】

コンピュータが直接解読して実行できる言語。

き ぐ う け ん さ
【奇偶検査 parity check】

2進符号において“1”の数を奇数あるいは偶数にするように、余分のビットを付加して2進符号の誤りの有無を検出すること。和による冗長検査の1つである。

き ぎ ょ う
【記号アドレス symbolic address】

記号を使って表わしたアドレス。コーディングの柔軟性を増し、変更を容易にするために用いられる。

記号アドレスによって書かれたプログラムを機械語に翻訳するのは、翻訳プログラムが行なう。

き ぎ ょ う げ ん ぎ
【記号言語 symbolic language】

ニーモニック・コードと記号アドレスで表現されるプログラミング言語。アセンブラ言語ともいう。

き ぎ ょ う
【記号プログラム symbolic program】

記号言語で書かれたプログラムで、名前や命令の特性や、プログラマに便利な記号などが、数字コードの代りに使われる。

ぎ じ め い れ い
【擬似命令 pseudo instruction】

人間にわかりやすい仮のコードで書かれたコンピュータ命令。言語翻訳プログラムで機械語に

翻訳される。

ぎじゆつぶん や
【技術分野 scientific application】

技術計算が主になるような情報処理の分野。

きじゆん
【基準アドレス (ベース・アドレス) base address】

1つのルーチンやデータのブロックに対して基準となるアドレス。これに相対アドレスを加えれば、絶対アドレスが得られる。

きじゆん
【基準レジスタ base register】

実行に先立って命令のアドレスを変更するために用いるレジスタ。

たとえば、命令の実行に先立ってオペランドのアドレスにそのレジスタの内容が加えられたり、減少させられたりする。

きじよう
【机上デバグ desk debug】

机上でプログラム中の誤りを見つけて直すこと。

きすう
【基数 radix】

てい
底ともいう。位取り記数法で、あるけたの重みにその数を掛けると、1けた上位のけたの重みが得られるような整数。

たとえば、10進数では各けたの基数は10であり、2進数では2、16進数では16である。

きほん
【基本カード master card】

(1) グループの最初にあって、そのグループに共通の情報を含んでいる紙カード。

(2) 変動が少ない基本的な項目がせん孔されている紙カード。

きほん
【基本ファイル master file】

(1) 変動の少ない基本的な項目から成るレコードでできているファイル。

(2) 処理結果などが蓄積されていて、次の処理の際、基本になるファイル。

▼ 【キャリッジ・テープ carriage tape】

ライン・プリンタの行送り制御のために使われる紙テープ。

きろくみつど
【記録密度 storage density】

記録媒体の単位長さ当りに記録できるビット数、字数、バイト数、語数。

けいえいじょうほう
【経営情報システム management information system】

MISともいう。経営者が決定を下すために必要なデータを提供する目的で設計されたシステムで、普通はオンラインで質問をすると、ただちにデータを得ることができる場合が多い。

けいさんこうもく
【計算項目 calculating field】

レコードの中の計算に使われる項目。

たとえば、送状の数量、単価、金額などの項目。

【けた上げ ^あ carry】

位取り記数法で表現されている2個の数のあるけたでの加算の結果が、基数に等しいか、基数を越した場合、1けた上のけたに1を加える操作、またはそのための信号。

【けたあふれ overflow】

演算の結果、取扱可能な範囲を越える計算結果を生ずる状態。通常、範囲を越えたけた数だけ、上位のけたが消えてしまう。

【けた移動(シフト) ^{いどう} shift】

1列に並んでいる数を右か左に移動させること。数が b を基数とする数字であるときに、 n けた右(または左)に移動することは、通常その数に b^{-n} (または b^n)を掛けることである。

【検孔 ^{けんこう} verify】

紙カードや紙テープにせん孔された内容について、誤りの有無を検査すること。

【検孔機 ^{けんこうき} verifier】

けん盤せん孔機でせん孔された紙カードや紙テープの内容の正確さを検査する機械。

【言語翻訳プログラム ^{げんごほんやく} language translator】

ある言語で書かれた命令を受取って、それと同じ意味の機械語命令や他の言語命令を作り出すプログラム。アセンブラ、コンパイラ、トランスレータ、ジェネレータなどに対する一般的な名称。

【検査数字 ^{けんさすうじ} check digit】

データを移動させるときに、数字が変化したら、その誤動作を発見できるようにするために、一定の規則で文字や語の中に付けておく、1つ以上の付加数字のこと。

【検査ビット ^{けんさ} check bit】

字やブロックなどの誤りの有無を検査するための付加ビット。

たとえば、奇偶検査ビット。

【原始プログラム ^{げんし} source program】

通常は記号言語や問題向き言語で書かれていて、言語翻訳プログラムによって機械語プログラムや他の言語プログラムに翻訳されるプログラム。

【けん盤 ^{ばん} key】

手動のカードせん孔機や紙テープせん孔機で、データをせん孔する機構。

ばん こうき
【けん盤せん孔機 (キーパンチ) key punch】

けん盤を手動で操作して、紙カードや紙テープに情報をせん孔する装置。

ご
【語 word】

1単位として取扱われる一連のビットまたは字の集り。

こうがく も じ
【光学文字 optical character】

光学文字読取装置で読取れるように設計された文字。紙テープ、紙カード、ジャーナル・テープなどに印刷された文字や、手書き文字などが含まれる。

こうがく も じ ようとりそう ち
【光学文字読取装置 (OCR) optical character reader】

一定の規則で印字された文字を、光学的に走査して読取る装置。

どうけいいんさつ
【合計印刷 (グループ印刷) group printing】

グループの見出しと合計だけを印刷すること。

こうじゆん
【降順 descending order】

逆順ともいう。コンピュータが持っている大きさの順序に従って、数字や文字の組合せが大きいものから小さいものへ順番に並ぶこと。

こうしん
【更新する update】

新しい情報や新しく取扱った内容などにもとづいて、基本的なファイルに変更を加えること。

こうもく
【項目 item】

(1) 一般に、ある組の中の要素。たとえば、レコードは、フィールドあるいはフィールドの集合という項目を含む。ファイルは、レコードという項目から構成される。表は、複数の記入項目 (entry) から構成される。

(2) 1単位として取扱われる、関連した字の集合。COBOLでは、処理の最小単位である。

ごかんせい
【互換性 compatibility】

ある装置で準備されたり、処理されたり、操作されたりしたデータを、そのまま別の装置で処理できる能力。

【コード coder】

(1) プログラムのコーディングをする人。

(2) エンコードの略。

ごちよう
【語長 word length】

1語の中に含まれるビット数または字数。

こてい きおく
【固定記憶 read-only storage】

自動的に書き込みができない記憶で、読出し専用に使われるもの。

通常、定数や常用ルーチンなどを記憶させるのに用いる。書き込み回路を切離した磁気ドラムなどがこの目的に用いられる。

こていしょうすうてん
【固定小数点 fixed point】

位取り記数法で数を表示する場合、左端（または右端）から数えて一定番目のけたの右に置く小数点のこと。

こていちよう
【固定長 fixed length】

ファイルが同一の長さのレコードで構成されていること。

こていちよう
【固定長レコード fixed length record】

ファイル内のすべてのレコードが、同一の長さであるようなレコード。

【コーディング coding】

- (1) データに該当するコードを付けること。
- (2) プログラミング言語を用いて、プログラムを記述すること。

【コード code】

データを記録したり、処理したりするときの負担を軽くするために、データを記号化して表現したもの。

【コード・ブック code book】

データとコードの対照表。

【コーナ・カット corner cut】

紙カードの種類や性質を見分けやすいように、紙カードの端を斜めに切ること。

こほる
【COBOL】

Common Business Oriented Languageの略。共通事務用言語。

【コマンド command】

- (1) プログラムを実行する際に、1ステップを開始させる1組の信号やパルス。
- (2) 入出力動作を指示する命令。入出力チャンネルで解釈され、実行される。
- (3) オペレーティング・システムのもとで、操作員が操作卓から指示を与えるときに使う命令。

【コントロール・システム control system】

プロセス・コントロール・システム。装置工業などで、装置の自動制御を行なうシステム。

【コンパイラ compiler】

自動プログラミングに使用されるルーチンの一種で、一般に人間にわかりやすい形（擬似コー

ド)で書かれたプログラムを、機械コード(または記号変換ルーチン、あるいは通訳ルーチンで処理ができる擬似コード)によるプログラムに翻訳するためのルーチン。

アセンブラとコンパイラとの相違は、前者では1つの擬似コードに1つの機械コードが対応するのに対して、後者では、1つの擬似コードから、一般にいくつかの機械コードが作り出されることにある。

通訳ルーチンとコンパイラとの相違は、前者では擬似コードから機械コードへの変換がプログラム実行の途中で、そのたびに行なわれるのに対して、後者では、変換がプログラムの実行に先立って事前に行なわれることにある。

【コンパイル compile】

問題向き言語を記号言語や機械語に翻訳すること。

【コンピュータ computer】

語源は、自動的に算術演算を行ってくれる機械の意。

現在では、電子計算組織のことをいう。電子計算機、電算機、計算機、EDPSなどともいう。

【再開(再起動) restart】

チェックポイントで記録されたデータを使って、プログラムの実行を再開すること。

【サイクル cycle】

1組の事象または現象が完了するまでの時間間隔。同じ順序で規則的に繰返される一連の操作。

【再配置可能 relocatable】

プログラムの中のアドレスを適当に調整して、ある記憶域から別の記憶域へ動かせば、別な領域でも実行できるようになるプログラムの仕組。

【作業域 working area】

プログラムで中間結果などを一時的に記憶しておくために使う領域。

【座席予約 seat reservation】

オンライン・リアルタイム処理の1つで、中央にコンピュータを置いて、通信回線を使い、航空機や列車の座席予約を行なうこと。

【サービス・プログラム service program】

コンピュータによる処理を援助するために使われる汎用的なプログラム。分類組合せプログラムやデータを転写したり複写したりするプログラムなどがある。

【サブルーチン subroutine】

ルーチンの一部分で、問題の一部を計算したり処理したりするための、それ自身でまとまった小ルーチン。

チン。

こう
【さん孔テープ punched tape】

データがせん孔された紙テープ。

さんじゆつしき
【算術式 arithmetic expression】

FORTRANなどで、加減乗除算を記述する式。

さんじゆつちんり えんざんそうち
【算術論理演算装置 arithmetic logical unit】

コンピュータの構成要素の1つ。算術演算、論理演算、けた移動などの操作を行なう回路を持つ。

さんしやうこうもく
【参照項目 reference field】

レコードの一部で、レコードの種類を識別したり、処理の仕方を判別したりするのに使われる項目。

じ
【字(キャラクタ) character】

数字、文字、特殊記号などをいう。

【ジェネレータ generator】

自動プログラミングに使用されるルーチンの一種で、必要なパラメータを与えられて、特定目的のルーチンを作り出すルーチン。

たとえば、報告書作成ルーチン、分類プログラム作成ルーチンなどがある。

一般的な翻訳ルーチンとジェネレータとの相違は、前者は汎用的な目的に使用されるものである。特定目的のルーチンを得るには、それに対応する問題向き言語や、記号言語で書かれたプログラムを与えなければならないのに対して、後者は応用範囲が限定されており、最小限必要なパラメータだけを与えれば、特定目的のルーチンを作り出すことができる点にある。

しかくけんさ
【視覚検査 sight checking】

せん孔した穴を光源にかざして見て、一群の紙カードが正しくせん孔されているかどうかを確かめること。

じき もじ
【磁気インク文字 magnetic ink character】

磁性材料を含む特殊なインクによって印字された文字。インクの粒子の存在は、磁気センサーで検出できる。

じき もじよとりそうち えむあいしーあーる
【磁気インク文字読取装置(M I O R) magnetic ink character reader】

磁気インクを使ってタイプされたり、書かれたり、印刷されたりしたデータを読取る装置。

【磁気カード じき きおく magnetic card】

磁気性の表面を持つカードで、その平らな表面を選択的に磁化して、データを記録することができる。

【磁気カード記憶 じき きおく magnetic card storage】

磁気カードの集りから成る記憶で、等速呼出しができる。磁気ディスクや磁気ドラムよりもアクセス時間は遅いが、記憶容量が大きい。

【磁気コア記憶 じき きおく magnetic core storage】

磁気コアの集りから成る記憶で、1個のコアは、1つの2進数字をたくわえることができる。通常、主記憶として使われる。

【磁気ディスク記憶 じき きおく magnetic disk storage】

回転ディスクの磁性材料の付いた表面に、情報を記録する記憶で、アクセス・アームによって回転ディスクの表面のデータを読み書きする。

【磁気テープ じき magnetic tape】

磁性材料がしみ込んだり、塗られたりしている物質から成るテープで、情報はその表面に、磁氣的に分極された点となって置かれることになる。

【磁気テープ記憶 じき きおく magnetic tape storage】

磁性材料で表面を覆ったテープの上に、情報を記録する記憶。

【磁気テープ装置 じき せうち magnetic tape unit】

磁気テープ記憶に情報を記録したり、情報を読出したりする装置。補助記憶装置、あるいは入出力装置として使用される。

【磁気ドラム記憶 じき きおく magnetic drum storage】

表面に磁性材料が塗られている回転円筒で、情報がその表面に、2進数で表現された小さな磁気点として記録されている記憶。

【磁気薄膜 じき はくまく magnetic thin-film】

磁性材料の層で、普通1ミクロンより薄く、しばしば論理素子または記憶素子として使われる。

【磁気フィルム記憶 じき きおく magnetic film storage】

磁気コアの代りに磁気薄膜を使って2進数を記録する記憶。主記憶として使われる。

【シーク時間 じかん seek time】

磁気ディスクや磁気ドラムなどのアクセス機構を、指定された場所に位置づける時間。

【資源 しげん resource】

処理のために必要な、データ処理システムの要素。オペレーティング・システムによって割当てられる。

たとえば、主記憶装置、入出力装置、中央演算処理装置、ファイル、プログラムなど。

【^{じす}JIS Japanese industrial standard】

日本工業規格。

通産省工業技術院では、日本工業標準調査会に専門委員会を常設して工業規格の標準化を進めている。コンピュータ関係では、コンピュータと情報処理システムに関する用語、問題の記述法、プログラミング言語、情報伝送上の諸特性などの規格を決める作業が進められている。

【システム system】

- (1) 一定の相互関連によって一体化され組織化された方法、手順、技術。
- (2) 特定の諸機能を果たするために必要な人間、機械、手法などの組織的な集合体。

【システム設計 ^{せつけい}system design】

システムの要件を分析して、新システムを設計すること。

【システム流れ図 ^{ながず}system flowchart】

データが、どのように流れ、どのような手順で処理されるかを書表わした流れ図。

【システム分析 ^{ぶんせき}system analysis】

どのような処理が必要か、その処理を行なうためにどのような操作が必要かなどを決めるために、対象となる活動、処理手順、処理方法、処理の技法、事務の内容などを検討すること。

【実行(ラン) ^{じつこう}execution, run】

1単位の処理を、最初から終りまで途切れることなく処理すること。

【実行サイクル ^{じつこう}execution cycle】

制御装置が命令を解釈してから、その実行を終るまでの動作のサイクル。

【実行プログラム ^{じつこう}executive program】

機械語でできており、主記憶にローディングすれば実行できるプログラム。

【実行用コンピュータ ^{じつこうよう}object computer】

目的プログラムを実行するコンピュータ。

【自動計算機 ^{じどうけいさんき}automatic computer】

人手によらないで算術演算や論理演算を実行できる計算機。現在ではコンピュータと同義語。

【自動データ処理 ^{じどうしよりえーで-びー}(ADP) automatic data processing】

自動計算機によるデータ処理。現在ではコンピュータによるデータ処理と同義。

じどう
【自動プログラミング automatic programming】

コンピュータを使って、記号言語や問題向き言語で書かれたプログラムから、機械語プログラムをつくること。

自動プログラミングの例としては、コンパイラ、ジェネレータ、通訳ルーチンなどがある。

しひょう
【指標(インデックス) index】

(1) 配列内の特定の要素を示すために用いられる記号や数。

たとえば、 $X_1, X_2, \dots, X_{100}$ で示される配列の各項目はインデックスとして、それぞれ 1, 2, $\dots$ 100 をもつ。

(2) ファイルや文書に関する見出しを順序よく記入し、その内容やあり場所を表示した表。索引ともいう。

しひょう
【指標レジスタ(インデックス・レジスタ) index register】

命令を実行する直前に、アドレス変更を行なうことができるコンピュータで、アドレス変更を用いる変更子を記憶しているレジスタ。

指標レジスタの内容は、条件付き飛越しの条件としても使われるのが普通である。

じむぶんや
【事務分野 business application】

事務処理や事務計算が主になるような情報処理の分野。

しゅうせきかい
【集積回路(IC) integrated circuit】

3~5mm²の半導体の板の上に、数十個の素子を埋込んで、整流・増幅・演算などの機能を持たせた回路素子の組。

しゅうだん こう
【集団せん孔 gang-punch】

先頭にある紙カードの情報の一部または全部を、後続する何枚かの紙カードにせん孔すること。

しんすう
【16進数 hexadecimal】

16を基数として表わされた数。

しゅきおく
【主記憶 main storage】

語、キャラクタ、バイトなどの単位でアドレスが付けられ、コンピュータ本体がそれを直接指定して、情報を書込んだり、読出したりできる内部記憶。

しんすう
【10進数 decimal】

10を基数として、数を表わす方法。

しゅつりょく
【出力 output】

(1) 処理されたデータ。

(2) ある装置からデータを取り出すのに使われる装置、または装置群。

(3) 内部記憶から外部記憶へデータを転送する処理。

しゅつりょくいき

【出力域 output area】

出力用に確保された記憶域。

しゅつりょくそうち

【出力装置 output unit】

コンピュータの構成部分の1つで、コンピュータから情報を出力する装置。

しゅつりょくばいたい

【出力媒体 output media】

コンピュータからデータを出力するために使う媒体。

しゅ

【主ルーチン main routine】

プログラムの骨幹になる部分。プログラムは、主ルーチンとサブルーチンとから構成されるのが普通である。

じゆんじ

【順次アクセス sequential access】

データを記憶装置から出し入れするとき、データが記憶されている物理的な順序に従って処理

して行くこと。

じゆんじしやり

【順次処理 sequential processing】

データを一定の順番で処理すること。

じゆんじよけんさ

【順序検査 sequence checking】

ファイルの中のレコードが、すべて昇順(あるいは降順)になっているかどうかの検査。

じゆんび

【準備ルーチン housekeeping routine】

コンピュータで主となる処理に先だって最初に実行されるルーチンの一部。初期設定をする。

じゆんへんせい

【順編成 sequential organization】

データが記録されている順序に従って、順番に処理されるように編成されていること。

じようい

【上位けた high-order position】

数字や語の左端のけた。

しょうきよ

【消去 erase】

記憶媒体上の情報を消去すること。

たとえば、磁気テープ上の状態を空白にすること。

じようけんつ ぶんき

【条件付き分岐 conditional branch】

指定した条件に従って、2つ以上のアドレスのうち、どちらか1つを選び、次の実行はそのアドレスにある命令から行なうように要求する命令。

しょうどうき

【照合機 collator】

組になったせん孔カードや、その他の文書を順番に照合したり、組合せをしたりする装置。

しょうどうじゆんじよ

【照合順序 collating sequence】

分類や照合を行なうときの約束された大小の順番。

しょうさいながず

【詳細流れ図 detail flowchart】

仕事の流れや、処理手順を詳細に書表わした流れ図。

しょうじゆん

【昇順 ascending order】

正順ともいう。数字や文字の組合せが、コンピュータが持っている大きさの順序に従って、小さいものから大きいものへ順番に並ぶこと。

じょうすう

【定数 constant】

値が変化しない量やデータ項目。

じょうほう

【情報 information】

一定の約束にもとづいてデータに与えた意味。

じょうほうけんさく

【情報検索 information retrieval】

記憶されているデータから、特定の方法と手段を用いて情報を捜し出すこと。

じょうほうしより

【情報処理 information processing】

与えられた情報から目的に沿った情報を得ること。通常のデータ処理のほか翻訳、図形処理、文字認識、音声の識別などが含まれる。

【ジョグル joggle】

せん孔カードの集りを、平らな表面に当てることによって、きちっとそろえること。

しよしき けいしき

【書式（形式） format】

記憶媒体や記憶内部あるいは印刷用紙などにおけるデータの配列の仕方。

【ジョブ job】

コンピュータで処理を行なうとき、操作員が指定する仕事の単位。

しより

【処理 process】

指定された結果を得るための、一連の系統的な動作。

【シリンダ cylinder】

磁気ディスクで一般に用いられている概念。磁気ディスク上で上下に重なり合っている1組のトラックをいう。

すいちよくけんさ

【垂直検査 vertical check】

媒体に記録された2進符号を、媒体の運動方向に対し垂直方向にビットを調べて検査すること。

【スイッチ switch】

- (1) 2種類の選択を行なうための回路や装置。
- (2) プログラムの中で2つの方向の動作が可能であり、どちらをとるかはプログラマが指定した条件によって決定されるような点。分岐点。

【水平検査 すいへいけんさ horizontal check】

媒体に記録された2進符号を、媒体の運動方向に対し水平方向のビットを調べて検査すること。

【スキップ skip】

- (1) 印刷の際、連続した用紙の印刷行の前後に空白行をつくること。
- (2) 一連の命令の中のある命令セットを無視すること。
- (3) データ処理で特定のデータを飛ばして処理すること。

【ステートメント statement】

- (1) 命令。命令を指示する動詞1つと、それに付随するオペランドやパラメータから成る。
- (2) FORTRANでは文のこと。

【スペース space】

- (1) データを記録するための記憶装置内の場所。
- (2) 1つ以上の空白。
- (3) 読取り位置や表示位置を、あらかじめ定められた形式に従って先へ進めること。

たとえば、印刷位置や表示位置を水平に右へ移したり、垂直に下へ移すこと。

【正規化 せいぎか normalize】

- (1) ある量をあらかじめ定められた範囲内におさめるように調整すること。
- (2) ある数の浮動小数点表示を、標準形式に置き換えること。すなわち仮数部があらかじめ定められた範囲内に入るように、指数部と仮数部を調整すること。

たとえば、仮数 x の範囲を $1 > |x| \geq 0.1$ に定めてあるコンピュータで、演算結果として 0.00632×10^8 が得られたときは、これを 0.632×10^8 に変更すること。

【制御 せいぎよ control】

- (1) コンピュータの構成要素である入力、記憶、演算装置、出力、制御という5つの基本的な機構の中の1つ。次の命令を選び、アドレスと命令部を解釈し、算術論理演算装置やその他の装置に指示を送り、命令を実行する。

- (2) 装置、データ、処理、概念などが一定の許された範囲内にあるように監視したり、調整し

たりすること。

(3) プログラムの中の一組の命令の実行順序を方向づけること。

せいぎよ
【制 御 カード control card】

プログラムに初期値を与えたり、プログラムの実行を制御したりするために使用する紙カード。

せいぎよそう ち
【制 御 装 置 control unit】

ディジタル型コンピュータの構成部分の1つで、命令を順次解読して必要な指令を与えて、自動的にコンピュータが処理を行なうように制御する装置。

せいぎよたく
【制 御 卓 console】

制御用キー、スイッチ、指示器などを備えており、操作員が必要に応じてコンピュータの動作に介入したり、これを監視したりするために設けられた卓。

せいぎよばん
【制 御 盤 control panel】

制御卓と同じ機能を備えた盤。

せいぎよよう も じ
【制 御 用 文 字 control character】

特殊な機能を持つ文字の一種で、特定の文脈の中でその文字がくると、制御操作が開始されたり、変更されたり、停止されたりするもの。

せい ど
【精 度 accuracy】

誤差が小さくなれば値が大きくなるような方法で表現される誤差の大きさの尺度。

【セクタ sector】

磁気ディスクや磁気ドラム装置のトラックの一部。通常、トラックは複数個のセクタから構成されている。セクタの代りにブロックということばを使うこともある。

【セグメント segment】

プログラムをいくつかの部分に分け、プログラム全体が一時に内部記憶装置に入らなくても、その実行ができるようにしたときの、それぞれのプログラムの部分。

ぜつたい
【絶 対 アドレス absolute address】

記憶場所を直接指定するアドレス。

よくせい
【ゼロ 抑 制 zero suppression】

数値を印字する場合、整数部の最上位の有効数字から左側のゼロを消すこと。

こう
【せん孔 (パンチ) punch】

紙カードや紙テープなどに、せん孔機で孔をあけること。

せんよう
【専 用 コンピュータ special purpose computer】

特定の目的のために設計されたコンピュータ。

【操作 (オペレーション) operation】

(1) コンピュータの1命令で指示された動作。たとえば、読み書き、加減乗除、移動、比較などの動作。

(2) 操作員 (オペレータ) がコンピュータを操作すること。

【相対アドレス relative address】

別に指定される基準アドレスを基準として、相対的に表わされたアドレス。

基準アドレスとしては、普通、ルーチンもしくはサブルーチンの第1語が記憶されるアドレス、またはその命令自身が記憶されるアドレスが用いられる。

【装置アドレス device address】

補助記憶装置や、入出力装置を示すアドレス。

【ソフトウェア software】

コンピュータの可能性を広げるために使われるプログラムやルーチンや技法。たとえば、コンパイラ、アセンブラ、各種のルーチンやサブルーチンなど。また、コンピュータと連結した技術文書の全部、たとえば、仕様書、回線ダイアグラムなどをいう。

【ゾーン zone】

特定の位置にあるビットの組合せで、文字、数字、記号を表わす場合と、主として文字、記号を表わす場合とがある。

図

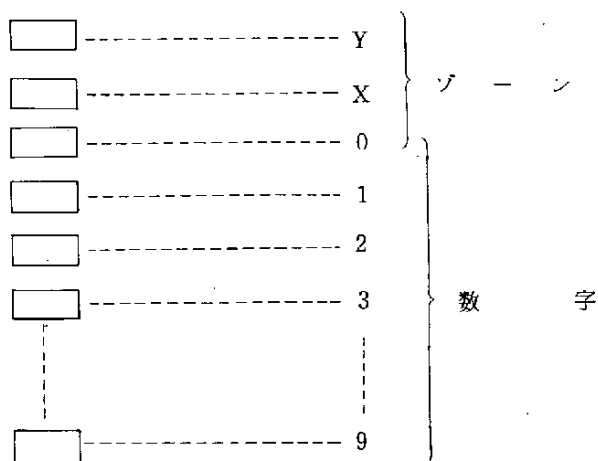
たとえば、80けたのカードでは、

通常、1けたに12個のせん孔位置があるが(図参照)、数字は0から9までのせん孔位置によって表示されるのに対し、文字、記号は上3個のせん孔位置と1から9までのせん孔位置との組合せによって表示される。この3個のせん孔位置がゾーンである。

【ゾーンせん孔 zone punch】

ゾーンにせん孔すること。

【ゾーン・ビット zone bit】



ゾーン部を表わすビット。8ビットで文字を表現するときの上位4ビット、6ビットで文字を表現するときの上位2ビット。

だい き ぼしゆうせきかい ろ
【大規模集積回路(LSI) large scale integration】

それぞれ機能を持った平位素子のアレイを数百個ほど論理回路的に相互接続させて作った集積回路。モノリシック集積回路を用いており、従来のICよりも複雑で高度である。LSIの開発によって従来よりもさらに原価を低減させ、性能を向上させ、信頼度を増した新しいシステムの構成が可能になった。

だとうせいけん さ
【妥当性検査 validity check】

あるコードが、許されているコードかどうかを確かめる検査。

【ターンアラウンド turnaround】

ジョブをコンピュータ室に提出してから、その結果を受取るまでの経過。

たんまつそう ち
【端末装置 terminal equipment】

データを入出力する装置。通常はオンライン・システムの入出力装置を指す。

【チェックポイント checkpoint】

プログラムの操作に誤りのないことが保障されている参照点。そのあとに誤りがあったときには、その参照点に戻って操作を再開することができる。

ちゆうおうえんざんしりそう ち
【中央演算処理装置 central processing unit】

命令の解釈や実行のために必要な制御回路や算術、論理、演算機構を含む装置。

ちようてい がた
【超小型コンピュータ mini computer】

主記憶が2,000ビット以上で、中央演算処理装置の価格が5,000千円以下のコンピュータ。

ちよくせつ
【直接アクセス direct access】

等速呼出しと同義語。

ちよくせつへんせい
【直接編成 direct organization】

任意の位置にあるレコードに直接アクセスできるように編成されたファイル。ファイル内に特別な索引は持たない。

ついせき
【追跡ルーチン tracer】

診断技法。プログラムの実行を分析して、出力装置に処理の過程を記録して行くルーチン。記録されるものは、命令とそのアドレス、オペランドとそのアドレス、処理の結果、使われた装置などが含まれる。

つきあわ
【突合せ matching】

2組のデータの特定の欄同士を比較して両者が同じであるかどうかを調べること。

ていあんしょ

【提案書 proposal】

システムの開発や、プログラムの開発に先立って作られる意見書。提案書が承認されると、開発の作業が始められる。

がた

けいすうがたけいさん き

【デジタル型コンピュータ（計数型計算機） digital computer】

数字表現で演算を行なうコンピュータ。

アナログ型コンピュータと対照される。

【ディスク・パック disk pack】

取りはずしのできる磁気ディスクの名称。

てきやうぎやうむ

【適用業務（アプリケーション） application】

コンピュータで処理される業務。

でぐち

【出口 exit】

ルーチンの出口。

【テスト・データ test data】

プログラムやルーチンが論理的に正しいかどうかを検査するときに使うデータ。

【データ data】

事象、概念、命令などを表現するもので、人間や機械による処理に適するように形式化されたもの。

【データ・アドレス data address】

データの記憶されている位置を示すアドレス。

しやうしやう

【データ収集 data collection】

1か所以上の地点から中央へ、データを集中すること。

しやうり

【データ処理 data processing】

必要な結果を得るためにデータの集計、加工、編集などを行なうこと。

つうしん

【データ通信 data communication】

通信回線を介して2つ以上の地点の間でデータを伝送すること。

てんそう じかん

【データ転送時間 data transfer time】

システムのある部分から他の部分へデータを送るのに要する時間。

【データ・ベース data base】

データの組織だてられた集合。

【データ 変換 プログラム へんかん data conversion program】

データの表現形式を、ある形式から別の形式に変更するプログラム。

【デバグ debug】

プログラムの中の誤りを見つけて直すこと。

【テープ・リール tape reel】

1巻のテープを巻いたもの。

【テンプレート template】

流れ図を書くための定規。

【等速呼出し (ランダム・アクセス) とうそくよび random access】

データを記憶装置から出し入れするときのアクセスに要する時間と、データの記憶位置との間に相関関係がない処理。

【等速呼出し記憶 とうそくよび random access storage きおく】

等速呼出しのできる記憶。

【特殊文字コード とくしゆもじ special character code】

英字、数字以外の文字コード。たとえば、*, \$, #など。

【閉じたサブルーチン と closed subroutine】

サブルーチンの一種で、主ルーチンのある点から制御を受けて、終ると主ルーチンの適当な点に制御を戻すように作られているルーチン。

【トラック track】

磁気ドラム、磁気ディスク、磁気テープなどの表面に一連の情報をたくわえる単位で、1個のヘッドで読出しまたは書き込みができる線状の部分。

【トランジスタ transistor】

真空管と同じ機能を果す小さな半導体素子。

【トランスレータ translator】

(1) 自動プログラミングに使用されるルーチンの一種で、一般に人間にわかりやすい形 (擬似コード) で書かれたプログラムを、機械語のプログラムに翻訳するためのルーチン。アセンブラ、コンパイラ、通訳ルーチン、ジェネレータなどがある。

(2) ある言語で書かれたプログラムを、別の言語のプログラムに変換するルーチン。

【取引カード とりひき transaction card】

新たに発生したり、変動したりした部分の情報を持つ紙カード。

基本カードと対^{つい}になることは。

【取引^{とりひき} ファイル transaction file】

変動的な性格や、一時的な性格を持っているデータ・ファイル。新たに発生したり、変動したりした部分の情報を持っており、基本ファイルを更新したり、基本ファイルと組合せて処理を行なったりするのに使う。

【流れ図^{ながず} flowchart】

コンピュータで処理を行なうのに必要な手順を図式化したもの。

【ナノ秒^{びよう} nanosecond】

1秒の10億分の1。

【なまデータ raw data】

処理されていないデータ。コンピュータに受入れられる形になっている場合とそうでない場合とがある。

【2進^{しんか}10進^{しん}コード (BCD) binary-coded decimal code】

10進数の各けたが1と0の組合せでできた2進数で表現されているコード。

たとえば、10進数の12は10位のけた1と1位のけた2のそれぞれが0001, 0010と表わされる。

【2進数^{しんすう} binary】

2進法に用いる数字。0, 1の2種が使われる。

【2倍精度^{ばいせいど} double precision】

コンピュータが本来取扱うことのできるけた数の、2倍のけた数を取扱うこと。

【ニーモニック・コード mnemonic code】

人間が覚えやすい形に簡略化したコード。コードは、基本的な働きがわかるようにつくられている。

たとえば、MPYは乗算、ACCは累算器を表わす。

【入出力制御システム^{にゅうしゅつりよくせいぎよ} (I O C S) ^{あいおーしーえす} input output control system】

入出力装置を制御するルーチンの集り、モニタやオペレーティング・システムのもとで働く。

【入出力制御装置^{にゅうしゅつりよくせいぎよそうち} input output control unit】

入出力装置を制御する装置。データ転送速度を調整したり、コード変換を行なったりする。

【入出力装置^{にゅうしゅつりよくそうち} input output unit】

入力装置と出力装置の総称。

にゅうしゅつりよく

【入出力チャンネル input output channel】

主記憶と、補助記憶や入出力装置の間にあって、データの転送を制御する装置。データ転送路やバッファを持っている。

にゅうりよく

【入力 input】

(1) コンピュータの内部記憶へ転送される情報。処理されるべきデータや、処理の制御を助けるための情報を含んでいる。

(2) ある装置へデータを記憶させるために使われる装置、または装置群。

(3) 外部記憶から内部記憶へデータを転送する処理。

にゅうりよくいき

【入力域 input area】

入力のために確保されている記憶域。

にゅうりよくそうち

【入力装置 input unit】

コンピュータの構成要素の1つで、情報を読み込む装置。

たとえば、カード読取装置や磁気テープ読取装置。

にゅうりよくばいたい

【入力媒体 input media】

コンピュータに情報を入力するときに使用する媒体。紙カード、紙テープなどがある。

はいせんばん

【配線盤 patch board】

機械の機能に融通性を持たせるために、機械の配線の一部分を多くの電気接点から成る盤に結線し、その盤上で短い接続線を用いて配線することができるようにしたもの。

配線盤は、印刷様式を指定したり、プログラムの一部機能を代行させたりすることに使用できる。通常、簡単に取替えができる。

【バイト byte】

1単位として操作される隣接した一連の2進数字。1バイトの大きさは、通常8ビットである。

がた

【ハイブリッド型コンピュータ hybrid computer】

アナログ型コンピュータとデジタル型コンピュータの能力を兼ね備えたコンピュータ。

はいれつ

【配列 array】

意味のある並び方をしている一連の項目。

【パック pack】

個々の項目を再現可能な方式で圧縮して記憶媒体に入れること。

【バックスペース backspace】

磁気テープを少なくとも物理的な1記録単位分だけ巻戻すこと。

【ハッシュ 合計 どうけい hash total】

ファイル中の各レコードについて、指定された項目の数字を合計したもの。磁気テープなどのデータを処理して行く途中で、主としてハードウェア的な誤りが発生しなかったかどうかを検査するために使われる。

【パッチ (切張り) きりば patch】

プログラムの誤りを改正したり、一連の処理手順を変更したりするために、プログラムに付け加えられるコーディング部分。

【パディング padding】

磁気テープなどで、特定の文字やパディング・レコードを使ってブロックをいっぱいにする技法。

【パディング・レコード padding record】

語、レコード、ブロック、ファイルなどのデータ単位を満たすために用いる、意味のないレコード。

【ハードウェア hardware】

コンピュータを構成する装置の総称。ソフトウェアの対照語。

【パラメータ parameter】

任意の値を与えてよいが、そのプログラムの中では、その値はずっと変らずにいるような変数。特定の目的や処理のために、一定の値が割当てられる変数、主ルーチンがサブルーチンに対して実行すべき処理の仕方や、実行に必要な変数などを与えるときに使われる。

【パリティ・ビット parity bit】

1組のビットの列に付加して、その組の中でONの状態にあるビットの総個数を、常に奇数または偶数に保つ働きをするビット。

【判断表 はんだんひょう decision table】

問題を処理するのに考慮しなければならない条件を、とるべき動作といっしょの組にして、1つの表にしたもの。

【パンチ・カード・システム punch card system】

紙カードを媒体にして、分類機、照合機、会計機などの機械と、いくつかの補助的な機械によって事務処理を行なうシステム。

【汎用 はんよう コンピュータ general purpose computer】

広い範囲の種々の問題を解くことができるコンピュータ。

はんよう

【汎用レジスタ general purpose register】

汎用的に使われるレジスタ。

びーえる わん

【P L/I Program Language I】

COBOL, FORTRAN, ALGOLの特徴を兼ね備えた高水準のプログラミング言語。

ひかく

【比較(する) compare】

2つのデータの大小関係や同一性を比べること。

ひき すう

【引数 argument】

独立変数の一種で、関数の値や演算の値がその変数の値によって決る。

たとえば、 $y = a \sin bx$, $y = ax^2 + bx + c$, $y = ae^x$ などの関数における x 。

【ビット bit】

情報の量の単位で、0と1、プラスとマイナス、onとoffなど2種類の情報を表わすことができる。2進数の1けた。

ひょう

【表(テーブル) table】

データの集合で、各項目が名前や、他の項目との相対的な位置関係などによって、一意に識別できるようになっているもの。

ひら

【開いたサブルーチン open subroutine】

サブルーチンの一種で、主ルーチンの一部として、主ルーチンの中に直接に組込まれるもの。閉じたサブルーチンと対照される。

【ファイル file】

コンピュータで1単位として取扱われる、関連したレコードの集り。

ほご

【ファイル保護リング file protection ring】

磁気テープ・リールに取付けるプラスチック製のリングで、テープに書込みを行なうときはこのリングを取付ける。書込み以外のときにはリングをはずしておけば、書込まれた情報が誤って消去されることがない。

【ファイル・ラベル file label】

磁気テープや磁気ディスクなどの媒体上に記録されているファイルを識別するために使われる特別なレコード。

ふぉーとらん

【FORTRAN】

Formula translationの略。

代数の記述法で表わすことのできる問題を記述するためのプログラミング言語で、主として、

技術計算用の問題向き言語。

ふくしや
【複写(する) duplicate】

カードや紙テープの内容をコピーすること。

ふごう いち
【符号の位置 sign position】

普通は数字の右端または左端にあり、その数の代数的な符号を表わす。

ふつきてん
【復帰点 return point】

(1) サブルーチンの中の主ルーチンへの戻り点。

(2) 印刷したり表示したりするとき、行の初めの文字の位置。

ぶつり
【物理レコード physical record】

主記憶と、磁気テープ記憶や、磁気ディスク記憶との間で、データを読み書きするときの物理的な単位。ブロックともいう。

ふどうしやううてん
【浮動小数点 floating-point】

数の表示法の一つで、小数点の位置を一定とせず、別に小数点の位置を指示する数を併記する表示方式。

たとえば、 -638020000 を -6.3802×10^8 または $-6.3802 E 8$ のように表示する。

ふどうしやううてん
【浮動小数点レジスタ floating-point register】

浮動小数点演算を行なうために使うレジスタ。

【ブラシ brush】

カード読取装置の一部で、せん孔された紙カードからデータを読取る機構の一つ。

【プログラマ programmer】

プログラミングをする人。

【プログラミング programming】

コンピュータのプログラムを作るには、計算処理の方式を定めて流れ図を作り、これをプログラム言語で書表わす。これらの作業の全部または一部をプログラミングという。

げんど
【プログラミング言語 programming language】

コンピュータのプログラムを記述するために用いられる言語。

【プログラミング・システム programming system】

特定のコンピュータのために用意されているプログラミング言語と言語翻訳プログラムの集り。最近では、サービス・プログラムも含めている。

【プログラム program】

コンピュータに必要な作業を指示するための手順を精密に記述したもの。

【プログラム・テスト program test】

プログラムの誤りを見つけるために、コンピュータにプログラムを実行させてみる。

【プログラム ^{ないぞうしき}内蔵式 stored program】

コンピュータの制御を、内部記憶に記憶されたプログラムで行なう方式。

【プログラム ^{ながず}流れ図 program flowchart】

プログラムに書く処理手順を記述した流れ図。

【プロシージャ procedure】

1 ステップずつ詳細に記述された問題解決のための手順。

【プロセス ^{せいぎよ}制御 process control】

コンピュータ（普通はアナログ型コンピュータ）を使って、操作や処理の自動調整を行なうシステム。

【プロセッサ processor】

(1) 原始プログラムを読取って翻訳や変換を行ない、目的プログラムを作り出す言語処理プログラム。言語プロセッサということもある。

(2) コンピュータの中央演算処理装置。

【ブロック block】

(1) 技術的あるいは論理的理由で、1単位として取扱われる一連の字、もしくは語の集り。

(2) 磁気テープや磁気ディスクに読み書きするときの物理的な単位。

【ブロック化 ^か blocking】

2つ以上のレコードを組合せて1つのブロックにすること。

【ブロック化 ^か 因数 ^{いんすう} blocking factor】

1 ブロック内に含まれているレコード数。

【ブロック ^{かんかんげき}間 ^{あいびー}間 ^い隙 (IBG) inter-block gap】

磁気テープや磁気ディスク上のブロックとブロックの間の間隔。ブロック間間隙には、データを読み書きできない。レコード間間隙 (IRG) ^{あいあーるじー}ともいう。

【分岐 ^{ぶんき} branch】

次の実行は、条件付きあるいは無条件に指定したアドレスから行なうように要求する命令。

【文書化 ^{ぶんしょか} (ドキュメンテーション) documentation】

情報を収集、編集して、記述すること。

ぶんぽうけん さ
【文 法 検 査 (シンタックス・チェック) syntax check】

言語の表現構造を、その言語の構造を決めている規則に従って検査すること。

ぶん るい
【分 類 sort】

データを特定の欄について一定順序になるように配列し直すこと。

たとえば、数の大きさの順、アルファベット順などに配列する。

ぶん るい とう もく
【分 類 項 目 sorting field】

レコードの中のレコードを分類したり、ませ合せたり、突合せたりするために使われる項目。

たとえば、送状番号、出荷番号、月日、得意先番号、商品番号などの項目。

【ヘッド head】

記憶媒体上のデータを読取ったり、消去したり、書込んだりするための機構。

たとえば、磁気ヘッド、光電式ヘッド、せん孔ヘッドなどがある。

へん かん
【変 換 conversion】

(1) データを、ある表現形式から他の表現形式に変えること。

(2) プログラムを、ある表現形式から他の表現形式に変えること。

へん しゅう
【編 集 (する) edit】

あとの操作のため、あるいは見やすくするためにデータの表現形式や配列を整えること。

たとえば、データの再配列、データの追加、不要のデータの削除、コード変換、ゼロ抑制などを行ったり、指定された形式で、データを表現したり、印刷したりすること。

へん すう
【変 数 variable】

ある集合内で値が変化する数。

ほ しゆ
【保 守 maintenance】

障害を排除して、ハードウェアやプログラムを良好に作動する状態に保っておくためのあらゆる活動。検査、測定、取替え、調整、修繕などが含まれる。

ほ じょき おく
【補 助 記 憶 auxiliary storage】

コンピュータによって制御され自動的に利用されるが、記憶場所の指定が語ごとではなく、語の集団に対して行なわれ、その集団を主記憶との間でやりとりすることにより、主記憶の記憶容量の不足を補うのに使用する記憶。磁気テープ、磁気ドラム、磁気ディスクなどがある。

ほ じょそう さ
【補 助 操 作 auxiliary operation】

中央演算処理装置の制御下でない装置によって実行される操作。

ほ すう
【補 数 complement】

基数を b とするとき、与えられた数から次の(1)または(2)に従って導かれる数。

(1) b に対する補数

与えられた数の各けたの数字を $b-1$ から引き、最下位に1を加えて必要ならば、けた上げを行なう。

例：2進数11010の2に対する補数は00110となる。10進数476の10に対する補数は524となる。

(2) $b-1$ に対する補数

与えられた数の各けたの数字を $b-1$ から引く。

例：2進数11010の1に対する補数は00101となる。10進数476の9に対する補数は523となる。

【ボリューム volume】

記憶媒体の1単位で、単一の読み書き機構でアクセスできるもの。磁気テープリール、磁気ディスクパックなど。

【ボリューム・ラベル volume label】

磁気テープリールや磁気ディスクパックなどのボリュームに記録された特殊なレコード。ボリュームを識別するための情報が含まれている。

【ホレリス・コード Hollerith code】

パンチ・カード式機械を考案したホレリスによって開発されたコード。

カード・コードとして使われている。

【^{ほんやく}翻訳(する) translate】

ある言語を他の言語に変換すること。

【^{ほんやくよう}翻訳用コンピュータ source computer】

コンパイルやアセンブルのために使うコンピュータ。

【マイクロフィルム・システム microfilm system】

コンピュータの主たる出力媒体に、マイクロフィルムを使い、マイクロフィルムにデータを記録し、保管するシステム。

【^{びよう}マイクロ秒 microsecond】

1秒の100万分の1。

【マーク・センス・カード mark sense card】

紙カード上の特定の位置に、電導性のペンで機械が読取れるマークを付けた紙カード。

【マクロ 命令 めいれい macro instruction】

同一の原始言語で書かれた一連のステートメントによって置き換えることができる原始言語の中の命令。

マクロ命令は、置き換えるステートメントの中のパラメータに対して値を指定することもできる。

【まぜ合せ あわ merge】

項目やレコードの順序を変更しないで、2つ以上の類似した順次ファイルを併合して、1つの順次ファイルにすること。

【丸め まる round-off】

指定されたけた数に収まるように、与えられた数に近似する数値を選ぶこと。丸めは四捨五入の規則（またはこれに相当するもの）によって行なわれることが多い。

【ミリ秒 びよう millisecond】

1秒の1000分の1。

【明細印刷 めいさいいんさつ detail printing】

集計されたものだけでなく、取引データも含めて印刷すること。

【命令 めい れい instruction】

機械に、演算その他一定の動作を指示することは。命令には、動作を指定する命令部と、1つ以上のアドレスから成るオペランドとが含まれているのが普通である。

【命令 アドレス めいれい instruction address】

命令語が記憶されている位置を示すアドレス。

【命令 コード めいれい operation code】

命令を表現するためのコード。命令は動作を指定する部分と、1個以上のアドレスを指定する部分とから成るが、命令コードは動作を指定するための表記形式である。

【命令 サイクル めいれい instruction cycle】

制御装置に命令が移り、命令を解読するまでの動作のサイクル。

【命令 セット めいれい instruction set】

ある特定のプログラミング言語で使用できる1組の命令。

【命令部 めいれい ぶ instruction part】

命令語のうち、命令コードの部分。

【命令 レジスタ めいれい instruction register】

実行する命令を入れておくレジスタ。

【メッセージ message】

(1) コンピュータと操作との間で交わされる情報。

(2) 始めと終りが明確に規定されたデータで、情報伝達を目的としたもの。

【メッセージ交換システム ^{こうかん} message switching system】

コンピュータに接続した通信回線を使って、メッセージを送受信するシステム。

【目的プログラム ^{もくてき} object program】

原始プログラムの翻訳処理の結果、出力されたプログラム。

【文字セット ^{もじ} character set】

字(キャラクタ)の順序集合。

たとえば、英語のアルファベットの26文字、ブール代数の0と1、モールス・コードの1組の信号、^{ゆーえすあすきー}USASCIIの128文字など。

【モジュール module】

装置や機構、部品、プログラム、ルーチンなどで、まとまった機能を持ち簡単に取替えができる単位。

【モニタ monitor】

システムの動作を観察し、監視し、制御し、確認する働きをするルーチンやハードウェアの機構。

【問題向き言語 ^{もんだいむ げんご} problem oriented language】

特定の種類の問題を、容易に表現できるように設計されたプログラミング言語。FORTRAN, COBOLなどがある。

【USASCIIコード ^{ゆーえすあすきー} USASCII code】

United States of America Standard Code for Information Interchangeの略。

7ビットの2進数コード。USAの標準コードとして設定されている。

【ユーティリティ・プログラム utility program】

汎用的なプログラムで、一般にデータ処理システムの操作を補助するために使われる。

【呼出し(コール) ^{よびだ} call】

主ルーチンから閉じたサブルーチンなどに制御を渡すこと。

【ライト・ペン light pen】

光を電気信号に変えるフォトセルが組込まれたペン。映像表示装置で使用する。

【ライブラリ library】

標準化されて一定の基準でまとめられたプログラム、ルーチンやサブルーチンの集り。

【ライン・プリンタ(行印字機) ^{ぎょういんじき} line printer】

各けたごとに独立した印字機構があり、横書きの1行を一度に印字できる印刷装置。

【リアル・タイム・システム real time system】

データが発生したとき、それをただちに処理するシステム。実時間システムともいう。

【リテラル literal】

原始プログラムにおいて、命令の中に書かれる文字列や数字で、それ自身がデータとなっているもの。

【リンケジ(連結) ^{れんけつ} linkage】

(1) 2つのルーチンを結び合わせる事、あるいはその手法。

(2) サブルーチンから戻るための命令とともに用いて、次に何を実行すべきかを決める命令やアドレスや変更子。

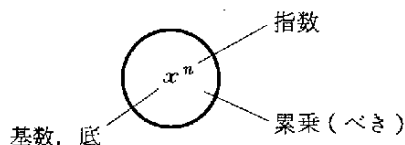
【累算器(アキュムレータ) ^{るいさんき} accumulator】

演算装置にある主要なレジスタで、算術演算、論理演算などの結果をたくわえる。多くの場合、累算器には1つの数値がたくわえられており、他から数値が入って来ると、両者の代数和でこれを置き換える。またたくわえられた数値について、けた送り、補数などの操作もできるのが普通である。

【累乗(べき) ^{るいじょう} power】

基数部と指数部とで表現された数値。

たとえば、1000を累乗表現にすると 10^3 。



【ルーチン routine】

あるまとまった機能を持ったプログラムの単位。

【ループ loop】

プログラムの中の一組の命令を繰返して実行すること。

【レコード record】

プログラムで1単位として取扱われる、関連した項目の集り。

【レコード間間隙 ^{かんかんげき} interrecord gap】

ブロック間間隙と同義語。

【レコード長 ^{ちよう} record length】

レコードの長さ。通常はワードや文字を単位として表わす。

【レジスタ register】

1語あるいは数語、ときには数けたの情報を記憶する装置で、特定の目的に使用され、随時、その内容を利用できるようになっているもの。

【レース・パンチ lace punch】

カードのあるけたの全せん孔位置にせん孔すること。

【ローダ loader】

実行可能なプログラムを外部記憶や補助記憶から読込んで、主記憶に記憶するためのルーチン。

【ローディング loading】

実行可能なプログラムを主記憶に記憶させること。

【論理判断 ^{ろんりはんだん} logical decision】

コンピュータの内部で行なう論理的な判断。同一性や大小関係などを比較して、結果をyesか、noの形で答える。

【論理命令 ^{ろんりめいれい} logical instruction】

入出力命令、算術命令、分岐命令以外の命令の総称。

【論理レコード ^{ろんり} logical record】

一般的にはレコードと同義。入出力処理の物理的な単位であるブロックを物理レコード (physical record) と呼ぶことがあり、それと対^{つい}になる用語。

物理レコードは、1つ以上の論理レコードから成っている。

【Yせん孔 ^{こう} Y-punch】

80欄カードの最上段のせん孔位置。

Xせん孔参照。

【ワード・マシン word machine】

内部記憶の記憶の単位や処理の単位が、語単位で行なわれるコンピュータ。

付録 4 教育施設・設備

目 次

I 教育施設・設備の計画と管理・運営	432
II 文部省通達による情報処理教育施設・設備基準	435
III 情報処理教育センター設置要項	451
IV 高等学校における教育用コンピュータについて	457
(財団法人日本経営情報開発協会の資料抜粋)	

I 教育施設・設備の計画と管理・運営

技術教育にとって実習室は、人間の形成と技術の習得とを、実践を通じて達成するための重要な場である。人間性は、環境によって大きく影響を受けるから、特に教育の場では、正しい態度・習慣などの基本的なしつけが自然に身につくような環境や、ふんい気を作るように条件を整備していかななくてはならない。整備された実習室では、技術者として必要な感覚や協同の精神を十分に養うことができるだろう。一般に実習室には、現用機器をそのままの形で教材としたもの、教材用に改良しくふうしたもの、実習特有の機能を持ったものなどがあるが、これらを有機的に組合せ、活用することがたいせつである。

また、実習室は実習指導の教育効果の点だけでなく、管理や運営を能率的に行なえるという点も考慮したものでなくてはならない。

実習室では、生徒が自主的に主体的に作業をすることが多いので、管理や運営の点からみると、予測しがたいような事態がいろいろ発生して、混乱することがある。そうした実習室の性格を十分考慮し、実状に応じた柔軟性のある内容にすることが必要である。施設・設備の計画とその管理・運営は、表裏一体であり、それがうまく調和しているかどうかによって、生徒の学習効果が非常に違ってくことになる。

以下、情報処理教育に関する施設・設備について実習室を中心に考えてみることにしたい。

それぞれの教育の場では、具体例を参考にして計画をたてていただきたい。なお、施設・設備の計画立案と管理・運営の方法とは、常に反省し、改善を加えなければならず、それらが停滞し、マンネリズムに陥ることのないように心がける必要がある。

1 施設・設備の計画

施設・設備を建設あるいは充実するに当たっては、各実習室で行なう実習内容をよく検討して、最も教育効果をあげ得るようにくふうしなければならない。ここでは、計画案の一例をあげる。

1.1 実習室の規模

実習室の規模は、学級数や学級定員、管理・運営の方法によって異なるが、生徒数を講義30名、実習10名とすると、1学級について次のような規模の実習室が必要である。

1	せん孔実習室	50 m ²
2	情報処理実習室	100 m ²
3	プログラミング実習室	150 m ²

1.2 各実習室の施設・設備計画

実習室の計画で留意する項目は、次のとおりである。

- ① 面積、高さ、床・壁・ドア・天井などの構造と材料、窓、向き。
- ② 採光、換気、しゃ光、通風、防湿、防じん、色彩、冷暖房。
- ③ 給排水、電源、照明、接地端子、アンテナ引出端子。
- ④ 準備室・管理室、実験台、戸などの大きさや位置。
- ⑤ 清掃、防火、防犯、その他の管理・運営に関する施設・設備。

次に各実習室について、計画の具体例を示しながら、各実習室で行なわれる実習の種別をあげ、設計・計画の留意点を考えてみよう。

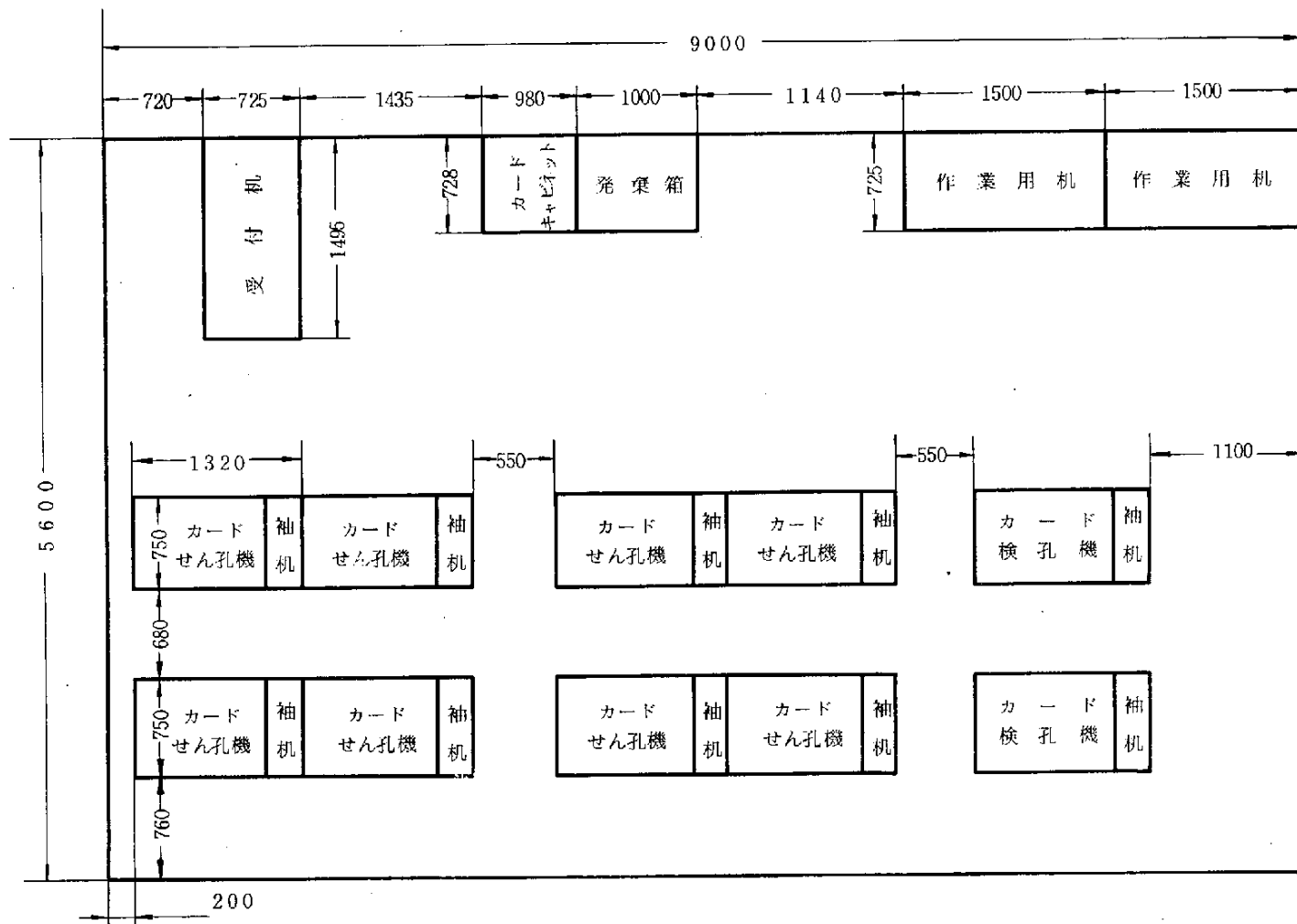
(1) せん孔実習室

この実習室では、カードせん孔機を使ってカードせん孔実習を行なう。

カードせん孔は、原始プログラムをカードにするときやプログラムをデバッグするときに必要なある。

次にせん孔実習室の例を示す。

せん孔実習室



せん孔実習室を計画するには、次の点に留意する。

(a) 騒 音

騒音は、作業者の耳の位置で、75フォン以下とすることが望ましい。

(b) 照 明

原票の位置の照度は、400ルクス以上とする。また、照明採光は、できるだけ明暗の差が少なく、しかもまぶしさを感じさせないような方法による。

(c) 室 温

室温は、冬期18℃を下らないように管理する。

(d) 作業室の広さ

せん孔機1台あたり4㎡以上とすることが望ましい。

(e) 作業姿勢等

いす、作業台および原票の読取りの難易等について人間工学的な見地から十分な配慮を払う。とくに適正な姿勢の保持に留意する。

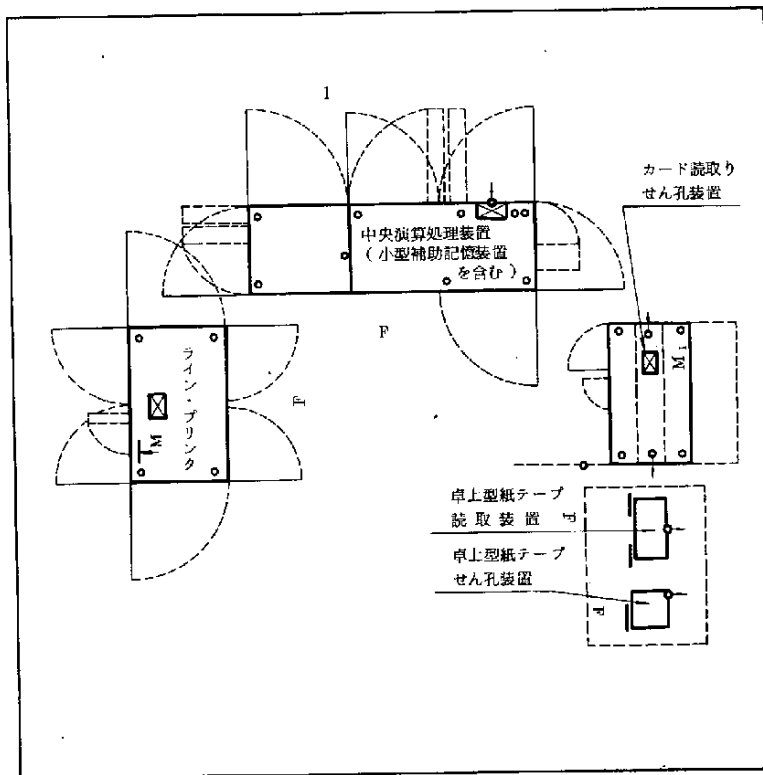
(2) 情報処理実習室

この実習室では、計算機を使ってプログラムの翻訳処理やプログラム・テストの実習を行なう。

情報処理実習室の中心は、電子計算機室である。その他に電源空調機室、保守サービス室、データ貯蔵室、デバグ室、管理事務室などが必要である。情報処理実習室とせん孔実習室の配置の例を示すと次のようになる。

電 源 空 調 室 (10.5㎡)	電 子 計 算 機 室 (36㎡)	デ ー タ 貯 蔵 室 (9㎡)
保 守 サービス 室 (10.5㎡)		せん 孔 実 習 室
管 理 事 務 室 (10㎡)	デ バ グ 室 (9㎡)	(15㎡)

次に電子計算機室の例を示す。



この実習室のうち、特に電子計算機室の計画に際しては、次のような点に留意する。

(a) 室の選定と構造

計算機室は、じんあい、熱、湿気を避けるために、最上階、1階以外の中間階に設けることが望ましく、直射日光の強い南向き、西向きはなるべく避ける。

天井の高さは、最低床上2.4 mとする。天井および壁面は、安定した空気調整と機器の騒音吸収のため、しゃ熱・吸音構造とすることが望ましい。

・窓は、空調効率を上げ、じんあいの侵入防止のために、できるだけエアタイト構造に近いもの（たとえばアルミサッシ）がよい。また直射日光を避けるためブラインドの設備が必要である。室機器の操作および保守のため、照度400ルクス（床上85 cm）以上にする。

床構造としては、床上げ（フリーアクセス方式、レースウェイ方式等の2重床構造のもの）が望ましい。

元床の強度は、 300 kg/m^2 以上を必要とする。

床表面材料は、清掃が容易で油に強く、じんあいの発生しにくいビニール系のタイルがよい。

室内の湿度が低下する冬季などには、摩擦によりかなり高電圧の静電気が発生し、放電による感電や計算機のエラー発生を招くおそれがあるので、使用するタイルは静電気の発生を抑えたものにする。

その他、保守用のコンセントの容量や位置、各装置の配置、機器の搬入口の大きさなど注意を要する点がいくつかあるので、個々の計算機の設置計画手引書を参照する必要がある。

(b) 空気調整条件

一般に計算機の空気調整条件は、室温 $15^{\circ}\text{C} \sim 30^{\circ}\text{C}$ 、湿度 $50 \sim 70\%$ の範囲内である。機器の発熱を考慮すると、ビルの冷暖房だけでは不十分で、専用の空気調整機が必要である。

通常、室温は、夏季 26°C 、冬季 21°C 、湿度は夏冬通して 55% 程度に設定する。室内の浮遊じんあいは種々のトラブルの原因となるので、じんあい量は 0.3 mg/m^3 以下に抑えることが望ましい。

(c) 電源設備

計算機を安定して実動させるためには、外来ノイズが少なく、しかも電圧変動が規定値 $\pm 10\%$ 、周波数変動が規定値 $\pm 1\%$ 以内の良質な電源の供給を必要とする。したがって電源事情がこの範囲を越える場合には、安定化のための配慮（たとえば自動電圧調整器：AVR等の設置）が必要である。そのために混触防止型の専用受電トランスを用いるか、電子計算機室内にノイズ除去のセパレート・トランスを使用する必要がある。電源の配管や配線、アースのとり方などについては、個々の計算機の設置計画手引書を参照する必要がある。

(3) プログラミング実習室

この実習室では、プログラミングの講義を受けるとともに、与えられた問題を分析して、流れ図を書き、コーディングするまでの過程の作業を行なう。プログラムのデバッグ、文書化の作業、プログラム完成後の反省と評価なども、プログラミング実習室で行なうことになる。

プログラミング実習室を計画するには、次の点に留意する。

(a) 部屋の大きさ

1人あたり 5 m^2 を確保したい。

(b) 作業空間

各種の文献・資料・用具を机上において、大判の用紙が使用できるような大型の机が必要である。またグループ編成にそなえて、机は移動可能なものでなければならない。

(c) 視聴覚教育用機器

オーバーヘッド・プロジェクタやスライド・プロジェクタが使えるような設備にすることが望ましい。

(d) 文書化と発表用器具

文書化と発表用に、黒板（あるいは白板）だけでなく、リコピー、イーゼル・ボードなどを備えておくことが望ましい。

2 施設・設備の管理と運営

実習室は、作業がいつでも円滑に実施できるような状態にしておく必要がある。

2.1 施設・設備の管理

2.1.1 管理の機構

管理がよく行なわれるためには、事務担当者の理解と協力が基礎となることはいうまでもないが、さらに事故の予防対策や整理・整頓などを円滑に実施するには生徒の協力が必要である。このことは、生徒指導上の問題としても重要なことである。

(1) 管理者の組織

実習室の管理責任者の所在を明確にしておかなければならない。

(2) 管理方式

備品や消耗品は、保管の難易を中心に考えると窓口を通して生徒に手渡すようにした方がよいが、学習効果から考えると、自由に生徒が取出して使用できる開架方式がよい。

前者は、備品や消耗品の出庫、入庫の場合に員数と故障の有無を点検することができる。しかも、その作業やその他の管理作業を実習題目として取上げることができれば、管理の本質を体験させることができる。後者の開架式は、生徒が

- ① 備品や消耗品の配置をよく知っている。
- ② 使用したものは完全に元に戻す。
- ③ 障害が生じたら必ず報告する。

などを実行する必要がある。この場合、生徒が気軽に使用できる便利さがあるが、反面、生徒の実践道徳の高揚と指導を合わせ行ない、管理にあやまちのないようにすることが必要である。

2.1.2 管理の実際

(1) 管理の内容

次のような項目についての管理が必要である。

- ① 備品、消耗品の使用
- ② 清掃
- ③ 電源

④ 盗 難

⑤ 火災防止

これらについては、毎月、日誌をつけるなどして、正確な記録を残したい。

(2) 検 査

管理の完全を期するためには、定期的な検査が必要である。

(3) 修繕・修理

補充備品は実習に供する間に故障することがかなりあるので、故障したらただちに報告させるようにしておく。

故障に対しては、すみやかな処置が必要である。消耗品は定期的に補充する。とくに電子計算機については、障害を詳細に記録し、障害統計をとることが必要である。このような表は、障害予防にも修理の際にも有用である。障害予防の方法としては「管理法の改善」と「実習指導の改善」などが考えられる。前者は、実習室内の機器の位置、通路の取り方、収容人員・指導人員、備品の運搬方法、格納場所などを研究することであり、後者は、備品使用の場合の指導上の問題で、生徒の技術・技能の程度に適した使用器具を選び、実験・実習の方法をくふうすることである。

2.2 施設・設備の運営

2.2.1 運営の計画

実習室の運営については、年間計画をたてる必要がある。人事関係から備品・消耗品などの準備計画まで、1年を通じて立案しなければならないものである。実習時期、実習時間に照らして、実習材料の購入、整備など年間計画ができなければならない。この点が運営上最もたいせつなところである。

2.2.2 実習室運営計画

実習室で行なわれる実習指導時間の表を作成するときは、準備、管理のための時間を併記明示して、運営に万全を期することが必要である。特に計算機使用の計画には、いちだんの注意が必要である。プログラム作成時間、カードせん孔の時間、計算機使用时间、デバグ時間などがうまく調和していないと、時間をむだに消費することになる。週間のうち特定の日だけ、一日じゅう通して実習指導を行ない、他の日は全然使用しないような時間編成は、なるべく避けるようにしたい。

以上、施設・設備の計画と管理・運営の概要について述べたが、それぞれの教育の場でくふうを

加え、独自の方式を生みだし、合理的な実施をはかることがたいせつである。

3 教育施設・設備の基準（参考例）

以下に教育施設・設備の基準について参考例をとりまとめて示してあるので、参照していただきたい。

3.1 施 設

施 設	床 面 積 (㎡)
せん孔実習室	5 0㎡
情報処理実習室	1 0 0㎡
プログラミング実習室	1 5 0㎡

3.2 設 備

(1) せん孔実習室

品目	数量	単 価 (円)	規格の標準その他
(せん孔用機械器具)			
カードせん孔機	8	1,500,000	英数字せん孔, 印刷機構つき
カード検孔機	2	1,500,000	
カード・ケージ	2	7,200	
(整理・ファイル用機械器具)			
カード・キャビネット	1	48,000	3,000枚×20組用
廃棄箱	1	5,000	不要カード廃棄用
(机・台・その他)			
わ き 机	10	7,200	せん孔機用 8 検孔機用 2
い す	10	3,000	
足 台	10	1,700	
作業用机	2	13,000	

(2) 情報処理実習室

品 目	数量	単 価 (円)	規格の標準その他
(データ処理機械器具)			
小型電子計算組織	1式	20,000,000 }	内部記憶 4K 語 補助記憶 32K 語 カード読取, せん孔装置 ラインプリンタ
30,000,000			
(整理・ファイル用機械器具)			
カード・キャビネット	3	48,000	操作手引書などのファイル用移動可能
ファイル・キャビネット	1	15,000	
帳票戸だな	2	24,000	
マニュアル・フォルダ	1	11,000	
データ連搬具	1	29,000	
廃棄箱	1	5,000	
(机・台・その他)			
テープパンチ	1	35,000	キャリッジ・テープ用
作業用机	2	13,000	
いす	1	3,000	操作卓用
げた箱	1	7,500	

(3) プログラミング実習室

品 目	数量	単 価 (円)	規格の標準その他
(講義用, 発表用器具)			
白板(または黒板)	2	30,000	黒板のときだけ必要
オーバヘッド・プロジェクタ	1	66,000	
スライド・プロジェクタ	1	57,000	
スクリーン	1	19,000	
イーゼル・ボード	2	18,000	
(文書整理用器具)			
ファイリング・キャビネット	2	15,000	
書だな	1	20,000	

(つづき)

品 目	数量	単 価 (円)	規格の標準その他
器具戸だな	2	59,000	
(机・台・その他)			
机・いす	10	机 23,000 いす 3,000	生徒用(3人用)
机・いす	1	机 7,000 いす 3,000	教師用

Ⅱ 文部省通達による情報処理教育施設・設備基準

文部省では、高等学校における情報処理教育を推進するため、下記の学科について産業教育実験実習施設・設備充実参考例を定めている（昭45・10・14付通知、高等学校の産業教育に関する学科における産業教育実験実習施設・設備充実参考例について）。

(1) 商業に関する学科

情報処理科およびこれに準ずる学科

(2) 工業に関する学科

情報技術科およびこれに準ずる学科

以下は、同参考例の内容および解説である。

1 情報処理科およびこれに準ずる学科の 施設・設備充実参考例について

文部省初等中等教育局
職 業 教 育 課

このたび、文部省において情報処理科およびこれに準ずる学科の実験実習用施設・設備充実参考例が決定され、この10月14日に発表の運びとなった。

商業に関する学科としては設備総額約4,000万円という、現行基準をはるかに上回る金額のものが始めて実現できて、関係者一同のひとしく喜びとするところである。各学校におかれても、その発表を一日千秋の思いで待ちあぐねてきたにちがいない。

この参考例に基づいて、さっそく本年より国庫補助が開始されるのであるが、明年、また明後年と、全国的に多数の情報処理科が設置される方向にあるときだけに、今後補助事業を通して商業高校における当該施設・設備が充実され、わが国における情報処理教育の飛躍的な発展のために役立つことができるならば、まことに幸いである。

(1) 情報処理科が必要とする施設・設備

情報処理科は、理科教育および産業教育審議会から出された情報処理教育推進についての建議に基づいて、改訂学習指導要領にも新しく位置づけられた学科であり、その目標にも述べられているように、電子計算機の利用に関する知識と技術を（かなり濃密に）習得させることが、本学科の現代にふさわしい特色となっている。そしてまた、産業教育におけるこれからの情報処理教育推進の役割をになって、大きな期待がかけられている学科でもある。

いうまでもなく、生徒に電子計算機の利用を学ばせるには、これに関する実験実習を十分に行なわせることがたいせつであり、このため、情報処理科では電子計算機を始めとする多額の設備を必要とする。こうした事情から、上記審議会の情報処理教育部会小委員会では、本学科における望ましい教育内容とともに、その実施に要する実験実習施設・設備についても慎重に検討して、その結論が小委員会報告の中に加えられた。今般決定を見た情報処理科の参考例は、この報告に従って原案が作成され、その後部分的な修正が施されたものであって、その内容は小型電子計算組織とせん孔タイプライタを中心とした設備構成となっており、これらを収容する4つの実習室によって成り立っている。

電子計算機は、事務科に見られるような超小型機では、本学科における特徴的な教育を円滑に進める上で機能的に不足すると判断され、2,000万円を上回る金額の小型機が考えられている。これには、多数の生徒による実験実習の利便を考慮して高速のリーダ、パンチ、ラインプリンタ等を含むとされている。入力媒体としては、わが国における小型機利用の実態に即して、また実験実習費の節約を図る意味もあって、紙テープを使用することになっている。たとえ情報処理科であっても、各学校にこれ以上の機械を配置することは無理であるので、必要に応じて、中型機を備えた情報処理教育センターを利用するようになる。

電子計算機に入力媒体を供給するインプット・マシンとしては、上述のとおり紙テープ・ペースを前提としたせん孔タイプライタが用意されている。その台数は、1学級45人の生徒数の3分の1にあたる15台が確保された。このほか、カードから紙テープに自動せん孔するマークカード変換装置も整備できるので、その活用が期待される。かなタイプライタは、せん孔タイプライタのキー・タッチ練習用としても役だつてあろう。

情報処理科の施設として、まず簿記実習室が掲げられているが、これは本学科における簿記会計教育のために必要とされるものである。プログラミングの机上実習を行なう施設は、「事務実践」を展開する施設と兼用になっているが、このプログラミング実習・事務実践室と、せん孔タイプライタを整備するせん孔実習室、および電子計算機を整備する情報処理実習室とは、後述するように情報処理教育を行なう上で一体として使用することになる。マークカード変換装置はプログラミング実習・事務実践室に置かれるようになっているので、「事務実践」におけるデータせん孔にも威力を発揮することであらう。

なお、せん孔実習室、情報処理実習室ともに十分な面積がとってあるので、準備室を設けることにも不自由しないはずである。

(2) 施設・設備の運用例

ここで、情報処理科における教育を、審議会答申に示された教育課程編成例に従って進めるように想定し、参考例による施設・設備のうち簿記実習室関係のものを除いて、その運用法を考えてみたい。

まず最初に、「事務機械」は1学級の生徒を15人ずつの3班編成で、ローテーションにより実習させたらと考える。第1班はかなタイプライタ、第2班はせん孔タイプライタ、そして第3班はさらに計算機とマークカード変換装置の2グループに分ける。したがって、実質は4つのグループ別に異なる機械の実習が展開されることになる。両タイプライタ班には、生徒各人が1台ずつの機械にあたることのできるのであるが、計算機械類は情報処理科では割愛されているので、

それらを他学科のもので間に合わせるなどの措置を講ずる必要があろう。

「電子計算機一般」で入力データを作成する実習は、主としてせん孔実習室で行なわれることになる。「経営数学」、「電子計算機一般」、「プログラミングⅠ」および「プログラミングⅡ」におけるプログラミングの実習では、まず1学級の全生徒をプログラミング実習・事務実践室に入れて、講義に続く机上での作業を開始する。時間の経過につれて、やがて進度の早い生徒から次々にコーディングを終了するから、教師は一応のチェックをして、これに通過した生徒を順次せん孔実習室に移動させる。そこでもまた、プログラムせん孔の終わった生徒を早い順に情報処理実習室に行かせて、電子計算機によるデバッグ、ランを行なわせ、これまでの作業結果の正誤を確認させる。これによってプログラムないしせん孔上の誤りを発見した生徒は、再びせん孔実習室に戻って修正を行なった上で、さらに電子計算機と取り組むことになる。たいていの場合、このような繰り返しを2度、3度あるいはそれ以上続けて、ようやく完全なプログラムができるといわれている。1つのプログラムを先んじて完成した生徒には、他の生徒が上記の段階まで追いつく間、プログラミング実習・事務実践室等において応用課題を解かせるようにするとよい。せん孔タイプライタ15台の構成は、決して3人で1本のプログラムを作成させることを前提として考えられたものではない。改訂学習指導要領にも示されているように、情報処理科では生徒ひとりひとりにプログラムを作らせることがたてまえであり、上述のような生徒の作業上の遅速を考慮して、せん孔タイプライタの台数がきめられているのである。学校設備では実習できない中型機の利用、カードによる入出力、外部記憶装置の使用などは、情報処理教育センターでの共同実習によって補完されるとよい。

さて、情報処理科における「事務実践」をどのように展開するかは、むずかしい問題である。現在、われわれにとってたいへん力強く感ずるのは、研究指定校の1つである小樽商業高校においてこのことに関する研究にすでに着手し、いよいよ試行段階にはいったということである。同校の試案によれば、45人の生徒を9人ずつ4企業と管理部に所属させて実践的活動を展開させるという計画で、この場合、せん孔実習室も使用することはもちろん、情報処理実習室に計算センター的な役割を果たさせるように考えられている。

同校の研究成果を大いに期待するものであるが、一般論としては企業の数をも4社に固定して考える必要はないと思う。

(3) 施設・設備の計画について

他の諸学科でもそうであるように、施設・設備は計画的に整備充実していくことが必要である。その計画作成に先だって、しっかりした教育計画が樹立されていなければならないことはもちろ

んであるが、あわせて十分な指導力を備えた担当教員を確保することが、きわめて重要である。これを援助する目的もあって、文部省では本年より期間3カ月の情報処理担当職員等養成講座を開始している。実習を重んずるため、受講者数の限られていることが難点であるが、今後講座数をふやす計画もあるので、できるだけ利用をおすすめしたい。

情報処理科にとって、電子計算機の設備は不可欠の要件であり、かつ、その早期導入が望まれる。また、電子計算機を利用する上で、数台のせん孔タイプライタが必要となろう。よって、少なくともこれらの設備と関係実習室とは、学科設置の初年度に整備されることが必要である。

施設・設備の整備にあたって留意すべき個々の問題につき多少ふれるならば、第1に、設置する電子計算機の種類によっては、情報処理実習室に空調装置が必要となるもようである。施設を計画する際、このことが忘れられてはならない。第2に、情報処理教育においてもその効果をあげるために、オーバヘッド・プロジェクタやビデオテープレコーダなどの視聴覚機器の使用が期待されるのであるが、これらの機器については、学校全体の設備計画の中で配慮できるとよい。第3の問題として、プログラミング実習・事務実践室の内部配置につき、この実習室がその名のとおり「事務実践」以外の諸科目におけるプログラミング実習にも使用されることを考慮して、設計にあたらなければならない点を指摘できる。一案として、事務実践用机・いす45組は想定される企業ごとに極端にまとめないで、すべて同一方向に整列させ、また、かなタイプライタを「事務機械」で実習させる都合上、室内の1カ所に集めて、必要によりアコーディオン・カーテンで仕切ることにしてはどうであろうか。

最後に、情報処理科の施設・設備は、情報処理教育のよりよき普及を図って校内他学科にもできるだけ利用させるよう配慮されなければならないと考える。さらにもし可能ならば、近隣他校の生徒にも一部開放されればと願っている。

高等学校の情報処理科およびこれに準ずる学科における産業教育実験実習施設・設備充実参考

例。

(1) 施 設

整理番号	施 設	床 面 積 (平方メートル)	備 考(施設の細分)
1	簿記実習室	9 0	情報処理実習室, 準備室 事務実践室, 準備室
2	せん孔実習室	1 3 5	
3	情報処理実習室	1 3 5	
4	プログラミング実習・ 事務実践室	2 2 5	
合 計		5 8 5	

(2) 設 備

1. 簿記実習室関係

整理番号	品目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
101	(簿記用器具) 簿記用帳簿式黒板	1組	15,000	15,000	15	机80×60×74cm
△102	(伝票整理用器具) ファイリング・キャビネット	1	25,000	25,000	20	
103	(机・台・戸だな類) 簿記机・いす	45組	4,500	202,500	15	
小計		基準細目金額		242,500円		
		原台帳細目金額		217,500円		

2. せん孔実習室関係

整理番号	品 目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
	(せん孔用機械器具)					
201	せん孔タイプライタ	15	1,000,000	15,000,000	8	リーダ付き
202	テープ・ホルダ	15	4,500	67,500	10	
	(整理・ファイル用機械器具)					
△203	テープ スプライサ	5	15,000	75,000	10	
204	テープ・ワインダ	3	8,500	25,500	10	
△205	紙テープ・キャビネット	1	30,000	30,000	20	
△206	ファイリング・キャビネット	2	15,000	30,000	20	
	(計算機械)					
△207	計 算 機	3	250,000	750,000	10	電動式または電子式、テンキー式またはフルキー式
208	(通信機械器具) 拡声装置	1式	61,000	61,000	10	
	(机・台・戸だな類)					
△209	作業用机	4	12,500	50,000	15	120×45×75 cm
△210	いす	15	2,000	30,000	15	
小 計		基準細目金額		16,119,000円		
		原台帳細目金額		15,154,000円		

3. 情報処理実習室関係

整理番号	品 目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
	(データ処理機械器具)					
301	小型電子計算組織	1式	22,000,000	22,000,000	8	高速紙テープ・リーダ、高速紙テープ・パンチ、高速印字装置などを含む。
	(整理・ファイル用機械器具)					
302	テープ・ワインダ	1	8,500	8,500	10	
△303	紙テープ・キャビネット	3	30,000	90,000	20	
△304	ファイリング・キャビネット	3	15,000	45,000	20	
△305	帳票戸だな	2	24,000	48,000	20	

整理番号	品 目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
△306	データ運搬具	1	22,500	22,500	10	
△307	(机・台・戸だな類) 作業用机	2	12,500	25,000	15	120×45×73cm
△308	い す	1	2,000	2,000	15	
		基準細目金額		22,241,000円		
		原台帳細目金額		22,008,500円		

4. プログラミング実習・事務実践室関係

整理番号	品 目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
	(プログラム用器具)					
401	プログラム用黒板	2	35,000	70,000	15	
402	帳票設計用黒板	1組	70,000	70,000	15	
	(タイプライタ)					
403	かなタイプライタ	15	80,000	1,200,000	8	手動式
	(データ処理機械器具)					
404	マークカード変換装置	1	1,900,000	1,900,000	8	マークカード読取り, 紙テープせん孔用
	(文書整理用器具)					
△405	ファイリング・キャビネット	2	25,000	50,000	20	
△406	ビジブル・キャビネット	2	35,000	70,000	20	
△407	ビジョン・ホール	2	5,000	10,000	15	100×120×40cm
	(証券作成機械器具)					
△408	チェック・ライタ	2	5,000	10,000	8	手動式
△409	印 鑑	1式	22,000	22,000	8	
	(通信機械器具)					
△410	室内電話機	8	11,000	88,000	10	相互式
	(机・台・戸だな類)					
△411	机・いす	45組	7,500	337,500	15	机 80×60×75cm 事務実践用

整理番号	品目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
△412	机・いす	11組	7,500	82,500	20	机75×60×75cm 計算機械器具用
△413	机・いす	15組	4,500	67,500	15	机80×60×67cm かなタイプライタ用
△414	カウンタ	8	13,000	104,000	15	172×30×110cm
△415	器具戸だな	2	20,000	40,000	20	180×90×45cm
小計		基準細目金額 4121,500円				
		原台帳細目金額 3240,000円				
合計		基準細目金額 42724,000円				
		原台帳細目金額 40,620,000円				

2 情報技術科およびこれに準ずる学科の 施設・設備参考例について

文部省初等中等教育局

職業教育課

(1) 基本構想

この参考例は、去る昭和44年12月3日に、理科教育および産業教育審議会からいただいた建議の中の小委員会報告に記された施設・設備に関する構想を基礎として作成されたものである。

情報技術科は、電子系コース、機械系コースおよびソフトウェア・コースの3つの類型を想定して構想されているもので、前記小委員会報告の資料Ⅱの施設・設備には、この3つのタイプのいずれの場合も含めて総括的に記されていたのであるが、このたび定められた参考例は、ソフトウェア・コース向きの施設・設備をもって、情報技術科の基準とすることになった。

その理由は、いずれのコースにおいても、電子計算組織は建議において指摘されている要件を満たさなければならぬいうに、電子系コースおよび機械系コースにあつては、そのほかにそれぞれのコースに固有の専門科目に関する施設・設備を設ける必要があること。この場合は、いずれも現行各学科一般の基準金額を大幅に越え、それらとのバランスがとれないこと。また、機械系コースおよび電子系コースは、実際上は、それぞれ機械科、電子科が既設されている学校に設置される場合が多いであろうし、その場合は、機械関係の設備は機械科と、電子関係の設備は電子科と共用することができるであろうと考えられたこと。

これらの理由から、情報技術科の基準はソフトウェア向きの施設・設備をもって構成されたのである。しかし上のような想定にもかかわらず、電子科、電気科をもたない学校で、電子系コースの情報技術科を設置する場合もありうるので、その場合の電子関係施設・設備については、別に考慮しなければならないのは明らかである。今後この問題はしかるべき措置を講ずる必要があると考える。

それはひとまずおくとして、以下参考例における施設・設備の個々について、重点的にふれてみたいと思うのであるが、参考例(別紙1)を参照しながらお読みいただきたい。

(2) 施設について

1のプログラミング実習室は、プログラム設計、フローチャートの作成、プログラム・コーディ

ング・シートの作成、およびこれらに付随する作業を行なう室である。この場合、1学級40名の生徒が同時に作業を行なうことができるように、また、用紙の大きさ、用具の配置等を考慮して、十分な作業空間が確保されるようになっている。

2のせん孔実習室は、プログラム実習室で作成したプログラム・コーディング・シートに基づき、計算機に読み込ませるためのせん孔テープを作成する室である。ここでは10人の生徒が同時にせん孔作業を行なえるように、また、デバギングも必要に応じて行なえるような作業空間が確保されている。

3のデータ処理実習室は、電子計算機室ともいえるもので、生徒は自作のテープやシステム・テープなどを計算機に読み込ませたり、定められた方法で計算機を操作して、目的の処理されたデータを入手するまでの実習を行なうのである。このほか、電子計算組織全体に関するアウトサイドの保守・管理などの実習も重要な内容の1つといえよう。

4の情報技術総合実習室は、1から3までの各実習室をとおして一連の情報処理技術の実習を行なうほか、情報技術科における教科・科目の内容に関する計算式の数値計算とか、1から3までの各実習室での情報処理は、主としてフォートランによる処理を想定しているから、この室では、計算機の言語機能に即して、アセンブラやマシン・ランゲージを使つての計算処理の実習を行なうことができる。また、計測機器を用いて、必要に応じて簡易な機能検査を行なえるように考慮されている。

これらの実習室で、生徒は、各科目の内容と有機的な関連のもとに、情報処理に関する一般的にいう初級程度以上の技術を習得することができると考えられている。

(3) 設備について

(A) プログラミング実習室関係の設備

この室は、前述のように広義に解したプログラミングの実習をする室であるので、プログラム設計のための各種資料や、それらを提示できる視聴覚教育用機器が特に望まれるのであるが、視聴覚教育用機器は、現行基準では、産業教育固有の設備ではないという観点から、いずれも設定されていないので、産振基準以外の今後の課題として留保され、今回は設けられないことになった。

プログラム設計の段階で、各種の文献・資料・用具を机上におき、大判の用紙も使用できるように、大型の机を用意し、ゆったりした作業空間が設定されたこと、また、プログラム・コーディング・シートの作成の際、フォートランあるいはアセンブラなどの言語に即して、定められた様式にしたがってコーディングしなければならないので、それを指導するための黒板が2式それぞれの様

式のものが用意されたことが、この室の設備の特色といえよう。

(B) せん孔実習室関係の設備

電子計算機のサイクル・タイムは、100万分の1秒〔 10^{-6} secまたは $1\mu s$ （マイクロセコンド）〕さらには10億分の1秒〔 10^{-9} secまたは $1ns$ （ナノセコンド）〕を単位として測られるような極小な時間であり、われわれの実感をもってはとうてい握できない速さとなっている。それにひきかえ人間が直接手を下す作業は、一般的にはおのずから限界があるので、教育用電子計算組織を構想するに当っては、特に両者の作業時間の調和をはかる必要がある。審議会小委員会においてもこれは特に強調された点であり、従来の汎用計算組織にはないせん孔部にオフラインのせん孔タイプライタを多数設置して、多人数を対象とした教育用の電子計算組織を構成したことは、今回の基準の特色といえる。これによって、従来1つのネックとなっていた、タイプライタの前の行列は解消されるはずであるばかりでなく、後述のようにプログラミング——せん孔——データ処理の一連の行程を、40名の生徒がスムーズに流れるように実習してゆけるはずである。

整理・ファイル用機械器具および什器類は、204 テープ・ウィングを除いて補助対象品目からはずされたが、これは従来の基準設定と同一観点からの措置で、従来どおりの取り扱いをしていただくことになるわけである。

(C) データ処理実習室関係の設備

情報処理教育に用いる電子計算機は、どのような方式で導入するのがよいかということが重要な課題であった。要するに産業教育のための設備費で導入するためには、レンタル制は不可であり、限られた予算のわくからはおのずから組織も限定されざるを得ないが、それらの条件をにらみ合わせて、多人数を同時に最も効果的に教育できる組織をという観点からこの基準が定められたことはすでに述べたところである。なかんずく中央処理装置およびその周辺機器の規模については、高等学校段階での学習内容・程度・範囲などを考慮し、情報処理教育センターとの相互補完の基本方針に照らして、一般的にいわれる小型級の中央処理装置と高速紙テープ・リーダ・高速紙テープ・パンチ、高速印字装置を含む組織が適当であるという結論が得られたのであった。また、主としてフォートランで処理する関係から、中央処理装置の記憶装置の記憶容量は、建議に述べられているように、コア・メモリ8KW以上またはコア・メモリ4KW＋ドラムその他のメモリが規格の標準とされている。高速紙テープ・リーダその他の入出力装置については、この基準細目品目の金額のわくの中で設置できる程度の規模のものであればよく、特に規格の明細を示さなかったのは、弾力的な運用を阻害しないように考慮したためである。

科学技術関係のデータ処理においては、単にデジタルな出力データを得るだけでなく、そのデジタル・データを迅速正確にグラフ化することがしばしば必要となる。そこで、電子計算機に直結して、直接グラフィック・データを得るようにXYプロッタが用意された。

また、情報処理教育センターは中型の電子計算組織をもって設備が構成され、テープおよびカードによる処理が行なえるので、学校における組織はテープ・ベースの組織に限定したのであったが、マーク・カードにはテープにない特性があるので、マーク・カード変換装置を別に設けることにした。これによって生徒には宿題を課し、実習室外でテープせん孔の過程までと同等の作業をマーク・カードで行なわせることにより、学習効果の向上とせん孔機の効率的な使用を保障することができる。

304のデータ通信実験装置は、今後のオンライン・システムの発達普及を考慮して、超小型電子計算機を端末機とし、これに変復調装置および模擬通信線を含めて構成されるものである。これは当面補助対象外品目の扱いとなったが、情報技術科の特色を裏付ける設備といえよう。

(D) 情報技術総合実習室関係の設備

401のカリキュレータは、電子式の計算機を一般的な呼称で品名を定めたもので、細目品目の金額のわくの中で設置することを原則とするのはいうまでもないが、情報処理技術に関する教育効果が十分期待できるものであればよいこととした。

1, 2の可能と思われる事例をあげるならば、1から3までの実習室で行なう実習とは独立して、アセンブラなどによる簡易または小容量のデータ処理や、後述の選択内容別の指導を実施できる超小型電子計算機を細目変更等により設けることができるであろう。また、卓上電子計算機で、ストアード・プログラムによる処理ができ、しかも、周辺にマーク・カード読取器、XYプロッタ、ブラウン管ディスプレイなどを接続した組織が考えられるが、このような形式のもので、科学技術系の計算処理を巧みに行なうものがあれば、これなどもよいであろう。その他各学校で、今後新しく開発される新機種の有効適切なものが得られるならば、それらも導入することができるようにした。

電子計測機器一式は、各室にある設備について、日常のアウトサイドの保守・管理に必要な手入れや必要に応じて簡易な機能検査に使用する計測機器で、規格の標準に示されたものを細目品目の金額のわくの中で設けられる適当な機能、精度、確度をもった品目をもって構成することになっている。

この室は第3学年の後半などで、選択内容ごとの班別実習を行なう場合に、独立してプログラミング——せん孔——データ処理の実習も行なえるように構想されているから、プログラム・コ-

ディングの指導に用いる黑板および机なども用意した。この机はプログラミングに使用するほか、電子計算機の出力データの事後検討や、電子計算機の機能の総合的な学習に供せられるよう配慮した。

(別紙1)

高等学校の情報技術科およびこれに準ずる学科における産業教育実験実習施設・設備充実参考例

(1) 施設

整理番号	施設	床面積 (平方メートル)	備考(施設の細分)
1	プログラミング実習室	180	データ処理実習室, 準備室
2	せん孔実習室	135	
3	データ処理実習室	135	
4	情報技術総合実習室	90	
合 計		540	

(2) 設備

1 プログラミング実習室関係

整理番号	品目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
101	(プログラム用器具) プログラム用黒板	2	35,000	70,000	15	120×45×73cm
△102	(机・台・戸だな類) 机	22	12,500	275,000	15	
△103	いす	40	2,000	80,000	15	
△104	戸だな	1	20,000	20,000	15	
小計		基準細目金額			445,000円	
		原台帳細目金額			70,000円	

2. せん孔実習室関係

整理番号	品目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
	(せん孔用機械器具)					
201	せん孔タイプライタ	10	1,000,000	10,000,000	8	リーダ付き
202	テープ・ホルダ	10	4,500	45,000	10	
	(整理・ファイル用機械器具)					
△203	テープ・スプライサダ	4	15,000	60,000	10	
204	テープ・ワイン	2	8,500	17,000	10	
△205	紙テープ・キャビネット	1	30,000	30,000	20	
△206	ファイリング・キャビネット	2	15,000	30,000	20	
	(机・台・戸だな類)					
△207	作業用机	4	12,500	50,000	15	120×45×73cm
△208	いす	10	2,000	20,000	15	
小計		基準細目金額 10,252,000円				
		原台帳細目金額 10,062,000円				

3. データ処理実習室関係

整理番号	品 目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
	(データ処理機械器具)					
301	小型電子計算組織	1式	22,000,000	22,000,000	8	高速紙テープ・リーダ、 高速紙テープ・パンチ、 高速印字装置などを 含む。
	(周辺機器)					
302	XYプロッタ	1	4,000,000	4,000,000	8	
303	マーク・カード変換装置	1	1,900,000	1,900,000	8	
△304	データ通信実験装置	1	8,500,000	8,500,000	8	
	(整理・ファイル用機械器具)					マーク・カード読取り、 紙テープせん孔用
△305	ファイリング・キャビネット	3	15,000	45,000	20	
△306	帳票戸だな	2	24,000	48,000	20	
△307	データ運搬具	1	22,500	22,500	10	

(つづき)

整理番号	品目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
308	テープ・ワイング	1	8,500	8,500	10	120×45×73cm
△309	紙テープ・キャビネット	3	30,000	90,000	20	
△310	作業用机	2	12,500	25,000	15	
△311	いす	2	2,000	4,000	15	
小計		基準細目金額			36,643,000円	
		原台帳細目金額			27,908,500円	

4 情報技術総合実習室関係

整理番号	品 目	数量	単価(円)	金額(円)	耐用年数	規格の標準その他
401	(計算機械器具) カリキュレータ	1	4,000,000	4,000,000	8	電子式
402	(計測機器) 電子計測機器	1式	2,000,000	2,000,000	8	バルボル, シンクロ スコープ, その他の 計測機器
△403	(製図用器具) 複写機	1	250,000	250,000	8	トランスベアレレンシ 作用
404	(プログラム用器具) プログラム用黒板	2	35,000	70,000	15	120×45×73cm
△405	(机・台・戸だな類) 机	22	12,500	275,000	15	
△406	いす	40	2,000	80,000	15	
△407	戸だな	1	20,000	20,000	15	
小 計		基準細目金額		6,695,000円		
		原台帳細目金額		6,070,000円		
合 計		基準細目金額		54,035,000円		
		原台帳細目金額		44,110,500円		

Ⅲ 情報処理教育センター設置要項

昭和45年11月4日付通知

1. 目 的

高等学校における電子計算機を中心とする情報処理教育（以下「情報処理教育」という）に係る生徒の共同実習、教員の研修等を行なうための施設として、情報処理教育センター（以下「センター」という）を設置し、都道府県における情報処理教育の推進を図る。

2. 設 置 者

センターは、都道府県が設置するものとする。

3. 性 格

センターは、地方教育行政の組織及び運営に関する法律（昭和31年法律第162号）第30条に規定する都道府県教育機関とする。

4. 組織および運営

- (1) センターに、所長、指導・研究担当職員、指導・研究補助職員、事務職員その他必要な職員を置くものとする。
- (2) これらの職員の待遇については、当該都道府県における同種の職にある職員と比べて均衡を失しないように措置するものとする。
- (3) センターの運営に当っては、学識経験者、高等学校教員、教育行政関係者等で構成する運営委員会を設け、運営の適正化および円滑化を図るものとする。
- (4) センターは、地域の実態に応じて、都道府県の総合的な教育研修センターの組織の一部として設置し、運営することができる。

5. 事 業

センターの行なう事業は、次のとおりとする。

- (1) 情報処理教育に係る生徒の実習を行なわせること。
- (2) 情報処理教育を担当する教員等に対する研修を行なうこと。
- (3) 情報処理教育に関する資料を収集し、整理し、保存し、および利用に供すること。
- (4) 情報処理教育に関する調査研究を行なうこと。
- (5) その他情報処理教育を推進するために必要な事業を行なうこと。

6. 生徒の実習

生徒の実習は、次の事項について、学校における学習の基礎の上に、地域の実態、学科の種類等に応じた適切な指導計画のもとに、行なうものとする。

- ア プログラミング
- イ 入力データのせん孔
- ウ データ処理
- エ 電子計算機の構成および機能
- オ 数値制御工作機械による作図および機械加工
- カ その他必要な事項

7. 教員等に対する研修

情報処理教育を担当する教員等に対する研修は、地域の実態、担当学科の種類等に応じた適当な計画のもとに、行なうものとする。

8. 施設および設備

施設および設備は、上記5の事業を実施するために必要なものを整備するものとし、その参考例を示せば、別紙のとおりである。

9. 国の助成

国は、別に定めるところにより、予算の範囲内において、センターの施設・設備の整備に要する経費について、その一部を負担する。

10. 指定都市（地方自治法（昭和22年法律第67号）第252条の19第1項に定めるものをいう）については、上記1から9までに定めるところに準じて取り扱うものとする。

別紙

施設・設備の参考例

1. 施設

施設名	面積(m ²)	施設名	面積(m ²)
電子計算機室	83	講義室(I)	90
プログラムせん孔室	83	同上(II)	90
データせん孔室(I)	83	研究・資料室	45
同上(II)	90	付帯施設	202
数値制御工作機械室	90	計	856

2. 設備

電子計算機室関係

品目	数量	単価(円)	金額(円)	備考
中型電子計算組織	1式	60,000,000	60,000,000	中央演算処理装置, カード入出力装置, 紙テープ入出力装置, ライン・プリンタ, 補助記憶装置, 制御卓, 入出力タイプライタなど。
空調調和装置	1式	1,350,000	1,350,000	
テープ・ワインダ	2	8,500	17,000	
カード・キャビネット	5	55,000	275,000	
紙テープキャビネット	3	30,000	90,000	
磁気テープキャビネット	1	71,500	71,500	
ファイリング・キャビネット	3	15,000	45,000	
帳票戸だな	2	24,000	48,000	
データ運搬具	1	22,500	22,500	
黒板	1	35,000	35,000	
作業用机	2	12,500	25,000	
いす	1	2,000	2,000	
小計			61,981,000	

プログラムせん孔室関係

品 目	数量	単 価(円)	金 額(円)	備 考
カードせん孔機	9	1,510,000	13,590,000	印字機構付き
カード・キャビネット	1	55,000	55,000	
ファイリング・キャビネット	2	15,000	30,000	
拡声装置	1式	56,000	56,000	
黒 板	1	35,000	35,000	
作業用机	6	125,000	750,000	
い す	18	2,000	36,000	
小 計			13,877,000	

データせん孔室(Ⅰ)関係

品 目	数量	単 価(円)	金 額(円)	備 考
せん孔タイプライタ	6	1,000,000	6,000,000	
テープ・ホルタ	6	4,500	27,000	
印刷電信装置	4	750,000	3,000,000	
テープ・スプライサ	4	15,000	60,000	
テープ・ワインダ	3	8,500	25,500	
紙テープ・キャビネット	1	30,000	30,000	
ファイリン・キャビネット	2	15,000	30,000	
拡声装置	1式	56,000	56,000	
黒 板	1	35,000	35,000	
い す	16	2,000	32,000	
小 計			9,295,500	

データせん孔室(Ⅱ)関係

品 目	数量	単 価(円)	金 額(円)	備 考
カードせん孔機	6	1,000,000	6,000,000	
カード検孔機	1	1,500,000	1,500,000	
分類機	1	1,800,000	1,800,000	
集団複写合計せん孔機	1	2,430,000	2,430,000	
翻訳機	1	2,490,000	2,490,000	

品 目	数量	単 価 (円)	金 額 (円)	備 考
カード・キャビネット	1	55,000	55,000	
ファイリング・キャビネット	2	15,000	30,000	
拡声装置	1式	56,000	56,000	
黒 板	1	35,000	35,000	
い す	14	2,000	28,000	
小 計			144,240.00	

数値制御工作機械室関係

品 目	数量	単 価 (円)	金 額 (円)	備 考
数値制御工作機械	1	10,000,000	10,000,000	制御用電子計算機付き
黒 板	1	35,000	35,000	
平 机	20	15,000	300,000	
い す	40	2,000	80,000	
小 計			10,415,000	

講義室 (I) 関係

品 目	数量	単 価 (円)	金 額 (円)	備 考
16ミリ映写機	1	270,000	270,000	
スライド映写機	1	65,000	65,000	
スクリーン	1	89,000	89,000	
ビデオ・テープ・レコーダ	1式	500,000	500,000	
テープ・レコーダ	2	40,000	80,000	
平 机	23	15,000	345,000	
い す	46	2,000	92,000	
講師用机・いす	1組	30,000	30,000	
小 計			1,471,000	

講義室（Ⅱ）関係

品 名	数量	単 価（円）	金 額（円）	備 考
プログラム用黒板	2	35,000	70,000	
帳票設計用黒板	2	35,000	70,000	
オーバヘッド・プロジェクタ	1	90,000	90,000	
TP作成機	1	199,000	199,000	
スクリーン	1	89,000	89,000	
平 机	23	15,000	345,000	
い す	46	2,000	92,000	
講師用机・いす	1組	30,000	30,000	
小 計			985,000	

研究・資料室関係

品 名	数量	単 価（円）	金 額（円）	備 考
和文タイプライタ	1	60,000	60,000	
英文タイプライタ	1	90,000	90,000	
かなタイプライタ	1	90,000	90,000	
ジアゾ複写機	1	198,000	198,000	
静電複写機	1	900,000	900,000	
謄写印刷機	1	250,000	250,000	
オフセット印刷機	1式	980,000	980,000	
自動原紙作成機	1	300,000	300,000	
図書戸だな	5	10,000	50,000	
ファイリング・キャビネット	5	25,000	125,000	
書 庫	2	15,000	30,000	
テープ・フィルム保管庫	2	30,000	60,000	
戸だな	2	20,000	40,000	
机・いす	3組	16,500	49,500	
平机	4	15,000	60,000	
いす	12	4,500	54,000	
小 計			3,336,500	
合 計			115,785,000	

Ⅳ 高等学校における教育用コンピュータについて

(日本経営情報開発協会教育問題研究会大学教育問題部会による提言

(昭和44年11月)から引用)

社会における情報処理技術者の不足および一般人の情報関係基礎知識の欠如を補うために、学校教育においてこれら情報科学に関する教育が広く実施されることが要望されており、大学においてはその重要性が認識され、着々と手が打たれつつあるが、それと対応して、最近では高等学校、特に商業工業の職業高等学校で、教育用コンピュータ導入の機運が高まっており、既に導入を果たした学校もかなりの数に上っている。

このような傾向は情報化社会への移行に対処するための教育態度としては喜ばしいことではあるが、教育指導理念を深く検討せず、安易にコンピュータを導入することは、教育効果上必ずしも好ましいことではない。殊に、最近では超小型機等比較的低廉な機器が出現し始めたので、その導入については、教育指導方針との関連において、慎重に検討する必要がある。

このような情勢に鑑み、当部会は高校におけるコンピュータ教育のあり方ならびに設置基準を明らかにすることが緊急に必要であると判断した。

高校におけるコンピュータ教育の指導要領といったものの作成には十分な研究と実験を必要とするが、そのためにもコンピュータの適切な導入が強く要請される。

また、コンピュータの将来の発達には、教育用として小型機を分散使用するか、あるいは端末機を通して大型機を時分割使用するか、研究すべき問題が多々あるが、ここでは小型機を分散配置する場合において、プログラミング教育に必要とされる性能、仕様はどのようなものが望ましいか、また将来端末機を使用する場合にも適用できることを配慮して、基準案を作成したもので、現在コンピュータ導入を意図している高校にとって一つの判断基準となることを期待している。

この提案は高校全部に小型機を設置することを意図するものではなく、主として職業高校を対象とし、かつ、当初は各地域ごとにモデル校を指定して、そこに導入し、それを当該地域の情報処理教育の核として逐次全地域に及ぼして行くことを想定している。

この結論によれば、小型機を分散配置する場合でも、1,500～2,000万円程度の機器を必要とする。しかしこれは決して500～600万円程度の超小型機がプログラミング教育にとって無意味であるというのではなく、そのような計算機を用いての教育には、機械の性能を補うに足りる

深い経験と知識をもった教員がいなければ成功を期しがたいと考えたからである。ただし、コンピュータの構造、作動原理に重点を置いた教育を行なう場合には、ミニコンピュータでもある程度の効果は期待できる。

また1,500～2,000万円程度の機器をあまり大量に導入することは、コンピュータの急速な発展途上においては陳腐化機器を大量に発生させる虞れがある。一方、今後ソフトウェアの改良によって、プログラミング教育が相当節約される可能性もあるので、設置に当ってはこの提案にあるごとく、融通性のある構成をとることが望ましい。そして各校がほぼ似かよった機器構成をとれば、ソフトウェアの開発経費は割安となるであろう。

なお、中心校への小型機の配置と並行して、共同利用センターの設置を促進し、ここに中型機を配置して、各高校に共同利用させるとともに、中心校の教育の補充や、教員の研修の場とすることが望ましい。

3.1 教育用小型コンピュータの標準仕様

ここでは小型機を分散配置する場合につき、

- (1) 当面、その小型機は最低限どのような性能と言語を具備する必要があるか、また、
- (2) 将来、どのような言語が開発されることが必要か。

を検討した。

3.1.1 ユーザたる学校側およびメーカーに対するアンケートの結果を総合するとき、当部会が想定した標準仕様は、教育用機器としては必ずしも満足すべきものではないと考えられる。

当部会が想定した標準仕様とは、4kワードのコア・メモリをもつ機械で、そのうち2kワードにコンパイラを常駐させ、残り2kワードでコンパイル・アンド・ゴーを行なうというものであり、したがって周辺に補助記憶装置をもたず、直接計算結果を取出す。すなわち、オブジェクト・プログラムをアウトプットする必要はなく、特別な入出力制御装置もいらぬもので、これならばかなり安価に製造しうるであろうし、高校の教育用としては適当であろうと判断したものである。

しかるに、アンケート結果から見ると、使用言語(FORTRAN)についてかなりの制限を加えなければならないし、このようなコンパイラの製作はむずかしい。さらに教育内容の発展可能性を考えると、上記のごとき機械の開発は技術的には可能であるが、教育効果上必ずしも有利ではないと思われる。

すなわち、

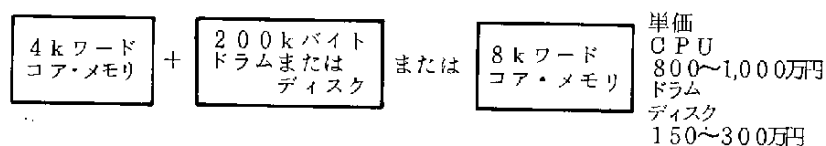
- ① かなり制限された言語しか使用できないので、実用という域には隔たりがある。
- ② コンパイラ的能力に発展性をもたせることができない。
- ③ 価格は1,000万円程度に達するであろう。

以上の点から、最低限どの程度の性能を具備する必要があるかを検討すると、次の2案が考えられる。

1つは、4kワードのコア・メモリと容量200kバイト程度のドラムまたはディスク(150～300万円)を使用し、JIS 3,000のFORTRANをそのまま使う。いま1つは、8kワードのコア・メモリで直接コンパイル・アンド・ゴーを行ない、FORTRANはJIS 3,000に若干の制限を加える。この両案の機器構成は下図のように考えられるが、価格は入出力装置を含めていずれの場合も1,500～2,000万円程度と見られる。しかし将来の発展性、卒業後の実用性等を考えると、教育効果は明らかに第1案の方が有利である。

構 成 図

C P U



I/O 装 置

コンソール・タイプライタ	1台(注1)	100~200
ライン・プリンタ	1台(注2)	500~800
カード	} リーダ	1台(注3)
紙テープ		
		100~500
		1,650~2,800
オフライン・パンチ	3台~10台(注4)	60~150

(注1) 不可欠のものではないが、あった方が教育上望ましい。

(注2) 能力は1行80～120字として、1分間に100～200行と想定。

(注3) 能力はカード・リーダの場合100～200枚/分、紙テープ・リーダの場合100～300字/秒と想定。

価格は紙テープ・リーダの方が安い、プログラム修正はカードの方がやりやすい。

(注4.)パンチは10台あれば1クラス(40人として)のプログラム・パンチが4時間程度で完了するので、10台程度が望ましいし、最低3台は必要である。1問題につき1人のプログラムをカード30枚程度とし、当初のパンチはミスが多いので、能率を30枚/時とみれば、所要時間は、

$$30 \text{ 枚} \times 40 \text{ 人} / 30 \text{ 枚 / 時} = 40 \text{ 時間}$$

∴10台あれば4時間でパンチができる。

次にコンピュータの教育用使用時間として、1台当り180時間(1カ月の標準稼働時間)は過大であろう。高校1学年の生徒数を200人とし、1人が在校中に2時間使用するとしても、月当りの使用時間は、

$$2 \text{ 時間} \times 200 \text{ 人} \times 3 \text{ 年} / 10 \text{ 月} \times 3 \text{ 年} = 40 \text{ 時間 / 月}$$

となる。したがってコンピュータを設置した場合には、機械の使用法、運用等については格段の工夫をこらす必要がある。

なお、初期教育については言語はやはりFORTRANを教えるのが本筋であろうから、その意味ではJIS3000をそのまま使える機械が便利である。それとともに別紙“制限条項”中の27番(logical IFの使用可能)および28番(欄記述子にAタイプを認める)の2項の機能を追加する。しかしながら第2案の方を採用するならば、上記“制限条項”中4番(データは整数形のみ)と5番(配列は1次元のみ)を採用する(それだけ制限条項が多くなる)。ただし27番と28番は第1案と同様追加するものとする。

以上の観点から、われわれは第1案の「4Kワードのコア・メモリ+200kバイトのドラムまたはディスク」という構成および上記の言語条件が、現在の標準仕様として最も適当であると判断する。

3.1.2

さて、(2)の言語の問題について論ずるならば、本来高等学校で教えるFORTRANは、JIS規格に捉われず、将来あるべき教科内容と関連づけて、JIS3000の仕様をさらに簡略化した言語を開発して、これについて作文と文法を徹底的に教育するのが1つの方向であろう。

このような言語を使用可能にするためには、現在のソフトウェア、ハードウェアについて、なお研究開発をすすめる余地が多分にあるものと思われる。そこで、前述の結論とは別個の問題として、このような認識が受入れられ、その結果、より適切なコンピュータ・システムと教育用プ

プログラム言語の研究が開始されることを期待する。

3.1.3

なお、これまではプログラム教育という面から考察をすすめたものであるが、特に工業高校のうち電子工学科などで、コンピュータの構造、作動原理に重点をおいたコンピュータ教育を行なう場合には、機械は前述の構成に捉われず、ミニコンピュータでもある程度の効果は期待できる。

以上の考察に基づき当部会は結論として、(1)の問題については第1案のごとき機器の構成を推奨するとともに、(2)の問題については早急に研究に着手することを要望する。

3.2 中古機の活用問題

ユーザーから返却された中古機を学校へ再配置することが、経済的な方法として考えられがちであるが、この考えは次のごとき難点がある。すなわち、

- ① 現在市場から返ってくるような機械ではJ I S規格のプログラム言語が使用できない。
- ② 返却された機械は、メーカーが自家用にするか、保守要員の訓練用に充てるので、学校に回す余裕がない。
- ③ 返却機はほとんど寿命がきているので、維持補修に莫大な費用がかかる。
- ④ メーカーはこれら旧式機の保守のために特別に要員を確保しておくことが困難であり、またそのような要員になり手が無い。

以上のことからプログラミング教育用として中古機を使うことは教育効果上はもちろん、経済面からみても得策でない。ただしハードウェアの教材としての利用価値は、大いに考えられる。

付録5 関係ある帳票様式

目 次

1 情報処理に関係ある帳票一覧	464
2 帳票様式のサンプル	465

1 情報処理に関係ある帳票一覧

主な帳票は次のとおりである（○印はサンプルを添付してあるもの）

(1) 文書（ドキュメント）に共通なもの

○(a) ドキュメント用紙

○(b) 表紙記入用紙

○(c) 中表紙記入用紙

(d) 目次記入用紙

(e) 参考資料記入用紙

(f) 用語登録簿

(2) 作業手順に関するもの

(a) 作業計画表

(b) 作業分担表

○(c) 作業指示書

○(d) 改訂・改版指示書

○(e) 作業完了報告書

(3) システムの仕様に関するもの

(a) システム仕様書

(b) システム流れ図用紙

(c) サブシステム仕様書

(d) 入力仕様書

(e) 出力仕様書

(f) ファイル仕様書

○(g) ファイル設計用紙

○(h) レポート設計用紙

(4) プログラムの仕様に関するもの

(a) プログラム説明書

○(b) プログラム流れ図用紙

○(c) テンプレート

○(d) コーディング・シート

○(e) テスト・データ記入用紙

(5) 検査に関するもの

○(a) 検査仕様書

(b) 検査成績書

(6) 操作に関するもの

(a) 操作説明書

(b) メッセージ一覧表

○(c) 操作指示書

○(d) キーパンチ指示書

(e) データ・ボリューム表

(f) プログラム一覧表

(g) 入出力一覧表

(h) ファイル一覧表

○(i) 磁気テープ外部表示ラベル

2. 帳票様式のサンプル

ドキュメント用紙

章
節
番
号
↓
 本
文
↓
 個
条
番
号
(i)
↓
 個
条
番
号
(a)
↓
 個
条
番
号
(i)
↓

	5	10	15	20	25	30
5						
10						
15						
20						
25						
30						
名 称			モジュール名			作 成 者
改版番号		改訂番号		改版改訂年月日	/ /	承認者
						承認年月日
					/ /	

表紙記入用紙

ドキュメント番号

一連番号

システム名
サブシステム名 } (作業名)

ドキュメント名(モジュール名)

副 題

作成年月

中表紙記入用紙

見 出 し 部

1 ドキュメント番号：

2 モジュール名：

3 作 成 年 月 日：

4 作 成 責 任 者：
 作 成 者：

5 改 訂 履 歴：

作業指示書

指示書番号

依頼書番号

納品先

日付

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
件名										使用機種										使用言語										プログラムステップ数																																																																					
設置・納品物件																																																																																																			

作業番号	所属	作業責任者
		氏名
作業時間	作業期間	年 月 日 ~ 年 月 日
単位	Y M	シニア
		ジュニア
		合計

作業担当者・作業内容・スケジュール

0	2	作業番号	所属	番号
---	---	------	----	----

作業者			作業内容とスケジュール(線表示)									
社員番号	氏名	作業時間(人月)										
11		17	20									
21		27	30									
31		37	40									
41		47	50									
51		57	60									
61		67	70									
71		77	80									

ドキュメンテーション スケジュール

0 3

計算機使用スケジュール

作業番号

所属

7 9

計算機名	準備時間(時分)	使用時間(時分)
10	14 35	18 30
23	27 28	31 32
36	40 41	44 45

記入要項

作業者のレベル(記入例)

- 5 既得のプロシージャ知識の経験有り
- 4 知識有り。スタディ不要
- 3 使用言語は知っている。プロジェクトは多少スタディ必要
- 2 使用言語は知っている。プロジェクトについてスタディ必要
- 1 使用言語は不安なし。プロジェクトについて知識なし
- 0 使用言語、プロジェクトとも知識なし

作業内容

- 6 スタディ
 - 1 デバッグ
 - 2 デザイン
 - 3 フローチャート
 - 4 コーディング
 - 5 マシンデバッグ
 - 6 コンポーネントテスト
 - 7 総合テスト
 - 8 検査
 - 9 その他
- ドキュメンテーション
- 例・・・プログラム構築
- 例・・・プログラム修正
- 例・・・プログラム解説
- 例・・・作成成書
- その他

0 4

作業番号

所 属

1 2

3 4 5 6

7 8 9

費 目		数 量		費 用 (千円)			備 考
				前 半 期	後 半 期	計	
直接材料費	磁気テープ	10	12	本			
	ディスクバック	13	15	個			
	カード	16	18	枚			
	印刷用紙	19	21	枚			
	インクリボン	22	24	本			
	その他	25	27				
直接経費	外注	計 算 機	前掲のとおり				
		カードパンチ	28	30	枚		
		その他	31	33			
	その他の経費	社内計算機	前掲のとおり				
		資料購入*	34	36	部		
		資料作成	37	39	部		
		印 刷	40	42	部		
		その他	43	45			
その他			46	48			
			49	51			
合 計 金 額		52				50	

* 市販の図書、雑誌は含めない。

改訂・改版指示書

第

年

月

日

ドキュメント番号		— —		作成者	
表 題				承認	
				保管場所	
改 版 ・ 改 訂	改版・改訂日	年 月 日	改版回数		改訂回数
	改訂場所				
	改版・改訂概要				
関 連 ド キ ュ メ ン ト	ドキュメント番号	表 題			
コ メ ン ト					
回覧先					
配布先					

作業指令書番号 _____
製品伝票番号 _____

昭和 年 月 日
報告書番号 _____

作業完了報告

作業番号 _____ 納品先名 _____
件 名 _____ 作業期間 _____
作業責任者 _____ 資料回収日 _____
作成者と担当割合等 _____

記述言語 _____ ステップ数 _____

作業の概要

特記事項，問題点など

保守について

担 当 者 _____

推定作業時間 _____ 推定計算機期間 _____

注 記

企画課記入欄

承認者

月 日

9VA-05-1

レポート設計用紙

名 称 _____

設計者 _____

日付 _____

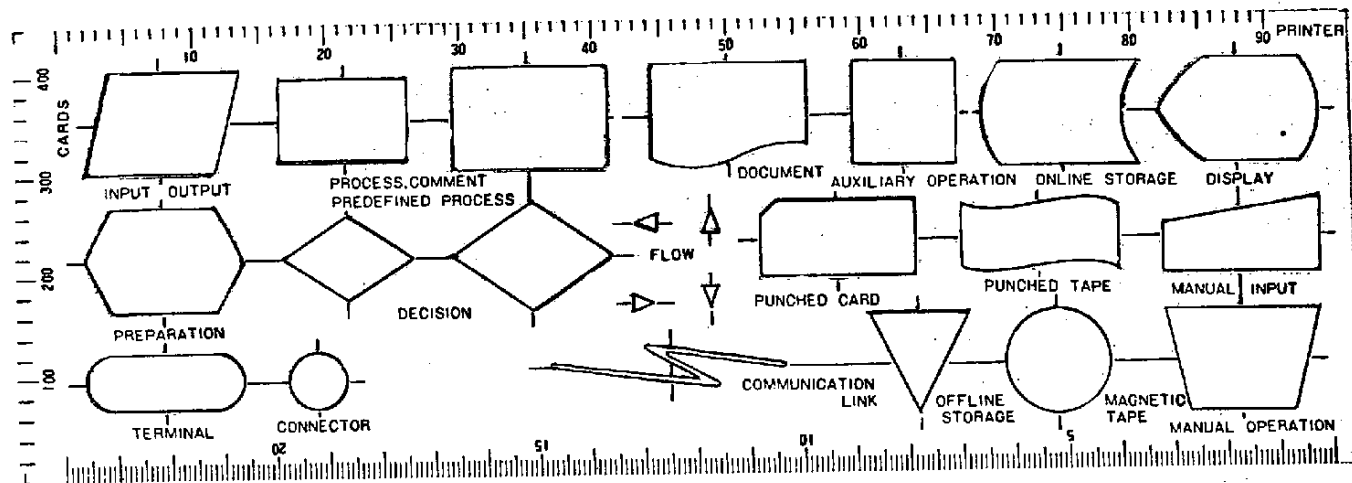
承認 _____

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	2	3	4	5	6	7	8	9	0	1	2	3	4
2	5	6	7	8	9	0	1	2	3	4	5	6	7	8
3	6	7	8	9	0	1	2	3	4	5	6	7	8	9
4	7	8	9	0	1	2	3	4	5	6	7	8	9	0
5	8	9	0	1	2	3	4	5	6	7	8	9	0	1
6	9	0	1	2	3	4	5	6	7	8	9	0	1	2
7	0	1	2	3	4	5	6	7	8	9	0	1	2	3
8	1	2	3	4	5	6	7	8	9	0	1	2	3	4
9	2	3	4	5	6	7	8	9	0	1	2	3	4	5
10	3	4	5	6	7	8	9	0	1	2	3	4	5	6
11	4	5	6	7	8	9	0	1	2	3	4	5	6	7
12	5	6	7	8	9	0	1	2	3	4	5	6	7	8
13	6	7	8	9	0	1	2	3	4	5	6	7	8	9
14	7	8	9	0	1	2	3	4	5	6	7	8	9	0
15	8	9	0	1	2	3	4	5	6	7	8	9	0	1
16	9	0	1	2	3	4	5	6	7	8	9	0	1	2
17	0	1	2	3	4	5	6	7	8	9	0	1	2	3
18	1	2	3	4	5	6	7	8	9	0	1	2	3	4
19	2	3	4	5	6	7	8	9	0	1	2	3	4	5
20	3	4	5	6	7	8	9	0	1	2	3	4	5	6
21	4	5	6	7	8	9	0	1	2	3	4	5	6	7
22	5	6	7	8	9	0	1	2	3	4	5	6	7	8
23	6	7	8	9	0	1	2	3	4	5	6	7	8	9
24	7	8	9	0	1	2	3	4	5	6	7	8	9	0
25	8	9	0	1	2	3	4	5	6	7	8	9	0	1
26	9	0	1	2	3	4	5	6	7	8	9	0	1	2
27	0	1	2	3	4	5	6	7	8	9	0	1	2	3
28	1	2	3	4	5	6	7	8	9	0	1	2	3	4
29	2	3	4	5	6	7	8	9	0	1	2	3	4	5
30	3	4	5	6	7	8	9	0	1	2	3	4	5	6
31	4	5	6	7	8	9	0	1	2	3	4	5	6	7
32	5	6	7	8	9	0	1	2	3	4	5	6	7	8
33	6	7	8	9	0	1	2	3	4	5	6	7	8	9
34	7	8	9	0	1	2	3	4	5	6	7	8	9	0
35	8	9	0	1	2	3	4	5	6	7	8	9	0	1
36	9	0	1	2	3	4	5	6	7	8	9	0	1	2
37	0	1	2	3	4	5	6	7	8	9	0	1	2	3
38	1	2	3	4	5	6	7	8	9	0	1	2	3	4
39	2	3	4	5	6	7	8	9	0	1	2	3	4	5
40	3	4	5	6	7	8	9	0	1	2	3	4	5	6
41	4	5	6	7	8	9	0	1	2	3	4	5	6	7
42	5	6	7	8	9	0	1	2	3	4	5	6	7	8
43	6	7	8	9	0	1	2	3	4	5	6	7	8	9
44	7	8	9	0	1	2	3	4	5	6	7	8	9	0
45	8	9	0	1	2	3	4	5	6	7	8	9	0	1
46	9	0	1	2	3	4	5	6	7	8	9	0	1	2
47	0	1	2	3	4	5	6	7	8	9	0	1	2	3
48	1	2	3	4	5	6	7	8	9	0	1	2	3	4
49	2	3	4	5	6	7	8	9	0	1	2	3	4	5
50	3	4	5	6	7	8	9	0	1	2	3	4	5	6
51	4	5	6	7	8	9	0	1	2	3	4	5	6	7
52	5	6	7	8	9	0	1	2	3	4	5	6	7	8
53	6	7	8	9	0	1	2	3	4	5	6	7	8	9
54	7	8	9	0	1	2	3	4	5	6	7	8	9	0
55	8	9	0	1	2	3	4	5	6	7	8	9	0	1

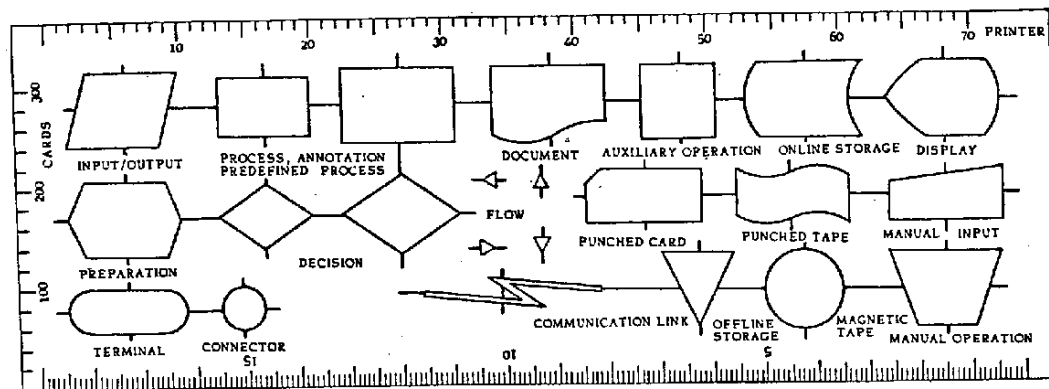
プログラム流れ図用紙

プログラム	プログラム番号	頁
図 番	図 名	プログラム名
		ページ

A1	A2	A3	A4	A5
B1	B2	B3	B4	B5
C1	C2	C3	C4	C5
D1	D2	D3	D4	D5
E1	E2	E3	E4	E5
F1	F2	F3	F4	F5
G1	G2	G3	G4	G5
H1	H2	H3	H4	H5
J1	J2	J3	J4	J5
K1	K2	K3	K4	K5



テンプレート (A 4 版用)



デ ー タ 名

テスト・データ 記入用紙

作成年月日

年

月

15

[illegible]

昭和 年 月 日

検査仕様書番号

検 査 仕 様 書

作 業 番 号

納 品 先

件 名

作 業 責 任 者

作 業 期 間

年

月

日

～

年

月

日

項 目	検査の有無	検査時期	検査者	検査方法 ㊦	検査実施日	検査 済印
planning						
functional specification						
system block chart						
detail specification						
manual external						
manual internal						
product test						
そ の 他						

㊦ 検査に使用する道具・検査のgrade等詳細は別紙(形式自由)に記入して添付する。

操作指示書

受付番号

IBM360-50 (I)		完了希望 日 時		受付日時 完了予定 日 時 分	
※ 指示項目 (申込者記入)				結果 (センター記入)	
作業者サイン					
① 処理内容	STANDARD	SPECIAL		① 完了	日 時 分
OS タイプ	MFT	PCP・MVT		② 完了状態 (システム側)	状態 コード ジョブ名
最大コアサイズ	100K・250K	K		・NORMAL	
予想全所要時間	10分以内	分		・ABEND	
打切時間	20分	分		・OP CANCEL	
JOB CLASS	A.B.C.J.KL	, ,		[説明]	
ターミナルコミュニケ	無	2260・2741			
立台希望	無	有			
PUBLICへの登録	無	有			
システム・ライブラリ登録	無	有			
② 入力カード	枚			③ 作業時間	分 秒
③ 出力カード	無	枚		通信欄 (申込者→担当者)	
SYSOUT CLASS	B	NOT USE			
UNIT 指定	NOT USE	00D			
カードの色	うぐいす色	色			
④ 使用言語	・ASM・PL/1・COB・RPG				
	・FORT・MPS・SSP・GPSS・GIS・PMS・LEMS				
⑤ 使用ユーティリティ	・PROGM・DASDR・MOVE				
⑥ JOB NAMES	1	2			
⑦ JOB 件数	件				
⑧ 印刷	UNIT	SYSOUT	UNIT 指定	フォーム 番号	トレイン カナ 数量
STANDARD	00E	A	禁	PN	禁・P 枚
SPECIAL	01E	01E	QN	KN	・P 枚
⑨ 私用ボリューム	VOL.SER.NO		UNIT	LABEL	
DISK				(注)TAPE の場合、必要ならばDCB等を記入	
TAPE			280	SL・NL	DSN
			281	SL・NL	DSN
***** (注1) 指示項目が STANDARD の場合、印以外は記入する必要はありません。 (注2) ※印の欄は必ず御記入ください。 印内に○をつけるか、又は指示項目を記入する。					
氏名	※	責任印	※	受付者署名	日 時
所属	※		※	室長	HEAD
会社	※	ユーザ番号		製品CHK	解禁CHK
				受付	

(受付番号)

キーパンチ指示書

年 月 日

ユーザー番号 	(会社名)	(所在地)	(電話)
(所属)	部	課	責任者印 氏名
No)	プリント 1する 2しない	使用コード 026 029	(数量)
使用言語 ASM FORT ALG RPG PL/I COB データ その他			
(希望納期) 年 月 日		(予定納期) 年 月 日	

磁気テープ外部表示ラベル

ファイル名	* 作成年月日
ファイル番号	* 保存期間
リール番号	* 消去予定日
作成者名	* 保管場所

付録 6 初級情報処理技術者教育時間配分表

	編	部	章	項 目	
				講 義	実 習
基礎編	200				
第1部 データ処理の基礎		50			
事務とデータ処理			15		
情報とデータ処理				3	
事務の機能				6	
データ処理の機能				6	
機械によるデータ処理			16		
データ処理方法改善の必要性				3	
データ処理方法の歴史				5	
機械による処理				8	
データ収集			19		
データ収集方法のいろいろ				3	
紙カード				4	
紙テープ				4	
その他のデータ収集媒体				5	
データの変換				3	
第2部 コンピュータのハードウェア		50			
ハードウェアのあらまし			10		
コンピュータの歴史				4	
現代のコンピュータ				3	
コンピュータ・システム				3	
入出力装置			14		
入出力装置の機能				6	
各種の入出力装置				8	
記憶装置			13		
データの表現				4	
内部記憶（記憶原理と各種記憶）				4	
補助記憶				3	

	編	部	章	項 目	
				講 義	実 習
外部記憶				2	
演算・制御装置			1 3		
演算の原理				4	
中央演算処理装置の基本要素				2	
制御機構				3	
算術論理演算機構				4	
第 3 部 コンピュータのソフトウェア		5 0			
プログラムのあらし			1 7		
プログラム記憶方式				4	
機械語				5	
プログラムの作り方				8	
プログラミング・システム			1 5		
プログラミング・システムの構成と用語				3	
サブルーチンとマクロ命令				3	
入出力制御システム				2	
サービス・プログラム				3	
プログラミング言語の種類				2	
ソフトウェアの発展				2	
言語のあらし			1 8		
アセンブラ				2	
FORTRAN				4	
ALGOL				1	
COBOL				5	
PL/I				4	
RPG				2	
第 4 部 コンピュータの利用		5 0			
システム設計			4 0		
システム設計の手順				7	

	編	部	章	項 目	
				講 義	実 習
コード設計				3	} 4
ファイル設計				6	
流れ図				5	
プログラム設計				5	7
情報処理規格				3	
コンピュータの応用			10		

	編	部	中項目	小項目	演 習
実習編	300				
第1部 プログラミング(FORTRAN)		220			
F O R T R A N 概 論			2		
F O R T R A N プログラムの形式			7		
コーディング・シートとコーディング・シート へ記入するときの注意				1	
F O R T R A N で使用できる文字				1	
行(注釈行, E N D 行, 開始行, 継続行)				1	
文				1	
文番号				1	
英字名				2	
データの型			10.5		
定数				2.5	
変数				3	
配列				5	
基本的な文(1)			18		
算術代入文				5	
G O T O 文と算術 I F 文				6	
入出力文(1)				7	
演習問題(1)					10

	編	部	中項目	小項目	演 習
基本的な文(2)			2 5. 5		
D I M E N S I O N 文				4	
D O 文と C O N T I N U E 文				7	
添字式				3	
S T O P 文と P A U S E 文				0. 5	
入出力文(2)				1 1	
演習問題(2)					1 0
基本外部関数と組込み関数			7		
演習問題(3)					1 3
文関数			7		
演習問題(4)					1 5
副プログラム			1 8		
関数 副プログラム				9	
サブルーチン副プログラム				9	
演習問題(5)					1 5
C O M M O N 文と E Q U I V A L E N C E 文			1 2		
C O M M O N 文				6	
E Q U I V A L E N C E 文				6	
演習問題(6)					2 0
書式なし入出力文			4		
補助入出力文			2		
演習問題(7)					2 0
まとめ			4		
第 2 部 プログラミング (COBOL)		2 2 0			
C O B O L 概論			1 0		
C O B O L の 歴 史				1	
C O B O L の 特 徴				1	
C O B O L の 言 語 体 系				3	
表記法				1	

	編	部	中項目	小項目	演 習
COBOLプログラムの構成				2	
COBOLプログラムの形式				2	
見出しディビジョンと環境ディビジョン			7		
見出しディビジョンの書き方				2	
環境ディビジョンの書き方				5	
演習問題(1)					10
データ・ディビジョン			28		
ファイルの構成				5	
データ・ディビジョンの書き方					
データ・ディビジョンの構成				3	
ファイル・セクションの書き方				15	
作業域セクションの書き方				5	
演習問題(2)					10
手続きディビジョン			55		
流れ図の復習				3	
手続きディビジョンの構成				2	5
文の種類と命令の種類				2	
入出力命令				7	
移送命令				6	
演算命令				8	
処理順序変更命令				8	
表の処理				10	5
演習問題(3)					40
条件命令				8	
注釈命令				0.5	
翻訳指示命令				0.5	
演習問題(4)					50
第3部 データ準備		30			
紙カード			30		

	編	部	中項目	小項目	演 算
紙カードの作成				1 5	
紙カードのチェック				1 0	
紙カードの取扱い上の注意				3	
紙カードの保管方法				2	
紙テープ					
紙テープの作成				必要に応じて便宜	
紙テープのチェック					
紙テープの取扱い上の注意					
紙テープの保管方法					
第4部 コンピュータの運用		5 0			
コンピュータの構成と 配置			4		
データの取扱い			8		
制御卓の操作			9		
入出力装置の操作			1 2		
操作の技法			9		
プログラムの保守			8		

索 引

凡例 基—基礎編

実—実習編

基、実に続く数字は

部、章の番号を表わす。

IDP (集中データ処理)	基1, 3
ISO	基4, 1
アクセス access	基2, 3
アクセス・アーム access arm	基2, 3
アクセス時間 access time	基2, 3
アセンブラ assembler	基3, 1
アSEMBル assemble	基3, 1
アドレス (番地) address	基2, 3
アドレス部 address part	基2, 4
アドレス変更 (修飾) address modification	基2, 4
アドレス・レジスタ address register	基2, 4
アナログ型コンピュータ analog computer	基2, 1
粗い流れ図 general flowchart	基4, 1
ALGOL	基3, 2
RPG	基3, 2
アンパック unpack	基2, 3
一括処理 (バッチ処理) batch process	基3, 2
EDP	基1, 3
EDPS	基1, 2
移動 (する) move	基2, 4
入口点 entry point	基3, 2
印刷装置 printer	基2, 2
インタプリタ (通訳ルーチン) interpreter	基3, 2
英数字コード alphanumeric code	基1, 3
映像表示装置 graphic display unit	基2, 2
Xせん孔 X-punch	基1, 3
エラー・メッセージ error message	実1, 実2
応用プログラム application program	基3, 2
オーバパンチ overpunch	実3
オフパンチ off-punch	実3
オフライン off-line	基2, 2

オペランド	operand	基2, 4
オペレーティング・システム	operating system	基3, 2
音声応答システム	audio response system	基2, 2
オンライン	on-line	基2, 2
オンライン・システム	on-line system	基4, 2
オンライン・バンキング・システム	on-line banking system	基4, 2
会計機	accounting machine	基1, 1
下位けた(単位けた)	low-order position	基2, 3
回転待ち時間	rotational delay time	基2, 3
解読器(デコーダ)	decoder	基2, 4
外部記憶	external storage	基2, 3
カウンタ(計数器)	counter	実2
拡張2進化10進コード(EBCDIC)	extended binary-coded decimal code	基2, 3
加算器	adder	基1, 1
仮数	mantissa	基2, 4
カード・イメージ	card image	基4, 1
カード送り	card feed	基2, 2
カード・ゲージ	card gage	実3
カードのけた(欄)	card column	基1, 3
カード・コード	card code	基1, 3
カード・システム	card system	基1, 2
カード・ジャム	card jam	実4
カード・スタッカ	card stacker	実4
カード・デッキ	card deck	実4
カード・ファイル	card file	実4
カード・ホッパ	card hopper	実4
カード読取せん孔装置	card reader punch	基2, 2
カード読取装置(カード・リーダ)	card reader	基1, 3
可変長	variable length	基1, 3
可変長レコード	variable length record	基4, 1
紙カード(カード)	paper card	基1, 1
紙テープ	paper tape	基1, 1
紙テープせん孔装置	paper tape punch	基2, 2
紙テープ読取装置	paper tape reader	基1, 3
緩衝域(バッファ)	buffer	基2, 2

監視プログラム	supervisor	基3, 2
関数	function	実1
記憶	storage	基2, 3
記憶装置	memory, storage unit	基2, 3
記憶ダンプ	storage dump	基3, 1
記憶容量	storage capacity	基2, 3
記憶レジスタ	storage register	基2, 4
機械語	machine language	基3, 1
奇偶検査	parity check	基1, 3
記号アドレス	symbolic address	基3, 2
記号言語	symbolic language	基3, 1
記号プログラム	symbolic program	基3, 1
擬似命令	pseudo instruction	基3, 2
技術分野	scientific application	基1, 1
基準アドレス(ベース・アドレス)	base address	基2, 3
基準レジスタ	base register	基2, 3
机上デバグ	desk debug	実1, 実2
基数	radix	基2, 4
基本カード	master card	基4, 1
基本ファイル	master file	基4, 1
キャリッジ・テープ	carriage tape	基2, 2
記録密度	storage density	基2, 2
経営情報システム	management information system	基4, 2
計算項目	calculating field	基4, 1
けた上げ	carry	基2, 4
けたあふれ	overflow	実1, 実2
けた移動(シフト)	shift	基2, 4
検孔	verify	実3
検孔機	verifier	基1, 1
言語翻訳プログラム	language translator	基3, 1
検査数字	check digit	基4, 1
検査ビット	check bit	基2, 3
原始プログラム	source program	基3, 1
けん盤	keyboard	実3
けん盤せん孔機(キーパンチ)	keypunch	基1, 1

語	word	基2, 3
光学文字	optical character	基1, 3
光学文字読取装置 (OCR)	optical character reader	基2, 2
合計印刷 (グループ印刷)	group printing	基4, 1
降順	descending order	基3, 2
更新する	update	実4
項目	item	実2
互換性	compatibility	基4, 1
コーダ	coder	基4, 1
語長	word length	基2, 3
固定記憶	read-only storage	基2, 3
固定小数点	fixed point	基2, 4
固定長	fixed length	基1, 3
固定長レコード	fixed length record	基4, 1
コーディング	coding	{ 基1, 1 基3, 1
コード	code	基1, 3
コード・ブック	code book	基4, 1
コーナー・カット	corner cut	基4, 1
COBOL		基3, 2
コマンド	command	基2, 4
コントロール・システム	control system	基4, 2
コンパイラ	compiler	基3, 1
コンパイル	compile	基3, 1
コンピュータ	computer	基1, 2
再開 (再起動)	restart	実4
サイクル	cycle	基2, 3
再配置可能	relocatable	基2, 3
作業域	working area	基3, 1
座席予約	seat reservation	基4, 2
サービス・プログラム	service program	基3, 2
サブルーチン	subroutine	基3, 1
さん孔テープ	punched tape	基1, 3
算術式	arithmetic expression	実1
算術論理演算装置	arithmetic logical unit	基2, 4
参照項目	reference field	基4, 1

字(キャラクタ)	character	基2, 3
ジェネレータ	generator	基3, 2
視覚検査	sight checking	実3
磁気インク文字	magnetic ink character	基1, 3
磁気インク文字読取装置	MICR	基2, 2
磁気カード	magnetic card	基2, 3
磁気カード記憶	magnetic card storage	基2, 3
磁気コア記憶	magnetic core storage	基2, 3
磁気ディスク記憶	magnetic disk storage	基2, 3
磁気テープ	magnetic tape	基1, 3
磁気テープ記憶	magnetic tape storage	基2, 3
磁気テープ装置	magnetic tape unit	基2, 2
磁気ドラム記憶	magnetic drum storage	基2, 3
磁気薄膜	magnetic thin-film	基2, 3
磁気フィルム記憶	magnetic film storage	基2, 3
シーク時間	seek time	基2, 3
資源	resource	基4, 1
JIS		基4, 1
システム	system	基4, 1
システム設計	system design	基4, 1
システム流れ図	system flowchart	基4, 1
システム分析	system analysis	実4, 1
実行(ラン)	execution, run	基4
実行サイクル	execution cycle	基2, 4
実行プログラム	execution program	基3, 1
実行用コンピュータ	object computer	基3, 1
自動計算機	automatic computer	基2, 1
自動データ処理(ADP)	automatic data processing	基1, 3
自動プログラミング	automatic programming	基3, 2
指標(インデックス)	index	基2, 4
指標レジスタ(インデックス・レジスタ)	index register	基2, 4
事務分野	business application	基1, 1
集積回路(IC)	integrated circuit	基2, 1
集団せん孔	gangpunch	実3
16進数	hexadecimal	基2, 4

主記憶	main storage	基2, 3
10進数	decimal	基2, 3
出力	output	基1, 2
出力域	output area	基3, 1
出力装置	output unit	基2, 2
出力媒体	output media	基1, 2
主ルーチン	main routine	基3, 2
順次アクセス	sequential access	基2, 3
順次処理	sequential processing	基2, 3
順序検査	sequence checking	実4
準備ルーチン	housekeeping routine	基3, 2
順編成	sequential organization	実2
上位けた	high-order position	基2, 3
消去	erase	基2, 2
条件付き分岐	conditional branch	基3, 1
照合機	collator	基1, 1
照合順序	collating sequence	基3, 2
詳細流れ図	detail flowchart	基4, 1
昇順	ascending order	基3, 2
定数	constant	実1, 実2
情報	information	基1, 1
情報検索	information retrieval	基4, 2
情報処理	information processing	基1, 1
ジョグル	joggle	実4
書式(形式)	format	実1, 実2
ジョブ	job	実4
処理	process	基1, 2
シリンダ	cylinder	基2, 3
垂直検査	vertical check	基2, 2
スイッチ	switch	基2, 3
水平検査	horizontal check	基2, 2
スキップ	skip	基2, 2
ステートメント(文)	statement	実1
ステートメント(命令)	statement	実2
スペース	space	実2, 2

正規化	normalize	基2, 4
制御	control	基1, 1
制御カード	control card	基3, 2
制御装置	control unit	基2, 4
制御卓	console	基2, 2
制御盤	control panel	基2, 2
制御用文字	control character	実2
精度	accuracy	基2, 1
セクタ	sector	基2, 3
セグメント	segment	基3, 1
絶対アドレス	absolute address	基2, 3
ゼロ抑制	zero suppression	基4, 1
せん孔(パンチ)	punch	基1, 2
専用コンピュータ	special purpose computer	基2, 1
操作(オペレーション)	operation	実4
相対アドレス	relative address	基2, 3
装置アドレス	device address	基3, 1
ソフトウェア	software	基2, 1
ゾーン	zone	基1, 3
ゾーンせん孔	zone punch	基1, 3
ゾーン・ビット	zone bit	基2, 3
大規模集積回路(LSI)	large scale integration	基2, 1
妥当性検査	validity check	基2, 2
ターンアラウンド	turnaround	基2, 2
端末装置	terminal equipment	基2, 2
チェックポイント	checkpoint	実4
中央演算処理装置	central processing unit	基2, 2
超小型コンピュータ	mini computer	基2, 1
直接アクセス	direct access	基2, 3
直接編成	direct organization	実2
追跡ルーチン	tracer	基3, 2
突合せ	matching	基1, 1
提案書	proposal	基4, 1
ディジタル型コンピュータ(計数型計算機)	digital computer	基2, 1
ディスク・パック	disk pack	基2, 3

適用業務 (アプリケーション)	application	基4, 1
出口	exit	基3, 2
テスト・データ	test data	実1, 実2
データ	data	基1, 1
データ・アドレス	data address	基3, 1
データ収集	data collection	基1, 3
データ処理	data processing	基1, 1
データ通信	data communication	基1, 1
データ転送時間	data transfer time	基2, 3
データ・ベース	data base	基4, 2
データ変換プログラム	data conversion program	基3, 2
デバグ	debug	基3, 1
テープ・リール	tape reel	基3, 2
テンプレート	template	基4, 1
等速呼出し (ランダム・アクセス)	random access	基2, 3
等速呼出し記憶	random access storage	基2, 3
特殊文字コード	special character code	基1, 3
閉じたサブルーチン	closed subroutine	基3, 2
トラック	track	基2, 3
トランジスタ	transistor	基2, 1
トランスレータ	translator	基3, 2
取引カード	transaction card	基4, 1
取引ファイル	transaction file	基4, 1
流れ図	flowchart	基3, 1
ナノ秒	nanosecond	基2, 3
なまデータ	raw data	基1, 1
2進化10進コード (BCD)	binary-coded decimal code	基2, 3
2進数	binary	基2, 3
2倍精度	double precision	基2, 4
ニーモニック・コード	mnemonic code	基3, 2
入出力制御システム (IOCS)		基3, 2
	input output control system	
入出力制御装置	input output control unit	基2, 2
入出力装置	input output unit	基2, 2
入出力チャネル	input output channel	基2, 2

入力	input	基1, 2
入力域	input area	基3, 1
入力装置	input unit	基2, 2
入力媒体	input media	基1, 2
配線盤	patch board	基1, 1
バイト	byte	基2, 3
ハイブリッド型コンピュータ	hybrid computer	基2, 1
配列	array	実1
パック	pack	基2, 3
バックスペース	backspace	基2, 2
ハッシュ合計	hash total	基4, 1
パッチ(切り張り)	patch	実4
パディング	padding	基4, 1
パディング・レコード	padding record	基4, 1
ハードウェア	hardware	基2, 1
パラメータ	parameter	基3, 2
パリティ・ビット	parity bit	基2, 3
判断表	decision table	基4, 1
パンチ・カード・システム	punch card system	基1, 1
汎用コンピュータ	general purpose computer	基2, 1
汎用レジスタ	general purpose register	基2, 4
PL/1		基3, 2
比較(する)	compare	基2, 4
引数	argument	実1
ビット	bit	基2, 3
表(テーブル)	table	実2
開いたサブルーチン	open subroutine	基3, 2
ファイル	file	基3, 1
ファイル保護リング	file protection ring	基2, 2
ファイル・ラベル	file label	基3, 2
FORTRAN		基3, 2
複写(する)	duplicate	基1, 1
符号の位置	sign position	基2, 4
復帰点	return point	基3, 2
物理レコード	physical record	基2, 2

浮動小数点	floating-point	基 2, 4
浮動小数点レジスタ	floating-point register	基 2, 4
ブラシ	brush	基 2, 2
プログラマ	programmer	基 4, 1
プログラミング	programming	基 3, 1
プログラミング言語	programming language	基 3, 1
プログラミング・システム	programming system	基 3, 2
プログラム	program	基 2, 1
プログラム・テスト	program test	基 3, 1
プログラム内蔵式	stored program	基 2, 1
プログラム流れ図	program flowchart	基 4, 1
プロシージャ	procedure	基 4, 1
プロセス制御	process control	基 2, 1
プロセッサ	processor	基 3, 1
ブロック	block	基 2, 2
ブロック化	blocking	基 3, 2
ブロック化因数	blocking factor	基 4, 1
ブロック間間隙 (IBG)	inter-block gap	基 1, 3
分岐	branch	基 2, 4
文書化 (ドキュメンテーション)	documentation	基 3, 1
文法検査 (シンタックス・チェック)	syntax check	基 3, 2
分類	sort	基 1, 1
分類項目	sorting field	基 4, 1
ヘッド	head	基 2, 2
変換	conversion	基 1, 1
編集 (する)	edit	基 2, 4
変数	variable	実 1
保守	maintenance	実 4
補助記憶	auxiliary storage	基 2, 3
補助操作	auxiliary operation	実 4
補数	complement	基 2, 4
ボリューム	volume	基 4, 1
ボリューム・ラベル	volume label	基 4, 1
ホレリス・コード	Hollerith code	基 1, 3
翻訳 (する)	translate	基 3, 1

翻訳用コンピュータ	source computer	基3, 1
マイクロフィルム・システム	microfilm system	基2, 2
マイクロ秒	microsecond	基2, 3
マーク・センス・カード	mark sense card	基1, 1
マクロ命令	macro instruction	基3, 1
まぜ合せ	merge	基3, 2
丸め	round-off	基3, 1
ミリ秒	millisecond	基2, 2
明細印刷	detail printing	基4, 1
命令	instruction	基2, 4
命令アドレス	instruction address	基3, 1
命令コード	instruction code	基3, 1
命令サイクル	instruction cycle	基2, 4
命令セット	instruction set	基3, 1
命令部	instruction part	基2, 4
命令レジスタ	instruction register	基2, 4
メッセージ	message	実4
メッセージ交換システム	message switching system	基4, 2
目的プログラム	object program	基3, 1
文字セット	character set	基2, 2
モジュール	module	基3, 1
モニタ	monitor	基3, 2
問題向き言語	problem oriented language	基3, 2
USASCIIコード	USASCII code	基2, 3
ユーティリティ・プログラム	utility program	基3, 2
呼出し(コール)	call	基3, 2
ライト・ペン	light pen	基2, 2
ライブラリ	library	実4
ライン・プリンタ(行印字機)	line printer	基2, 2
リアル・タイム・システム	real time system	基4, 2
リテラル	literal	実2
リンケジ(連結)	linkage	基3, 2
累算器(アキュムレータ)	accumulator	基2, 4
累乗(べき)	power	基2, 4
ルーチン	routine	基4, 1

ループ	loop	実1, 実2
レコード	record	基4, 1
レコード間隙	interrecord gap	基2, 2
レコード長	record length	基4, 1
レジスタ	register	基2, 4
レース・パンチ	lace punch	実3
ローダ	loader	基3, 2
ローディング	loading	基3, 1
論理判断	logical decision	基3, 1
論理命	logical instruction	基2, 4
論理レコード	logical record	基4, 1
Yせん孔	Y-punch	基1, 3
ワード・マシン	word machine	基2, 3

—— 禁 無 断 転 載 ——

昭和 46 年 3 月 発 行

発行所 財団法人 日本情報処理開発センター

東京都港区芝公園21号地1番5

機 械 振 興 会 館 内

TEL (434) 8 2 1 1 (代表)

印刷所 三協印刷株式会社

東京都渋谷区渋谷3-11-11

TEL (407) 7 3 1 6

