

FUJITSU Software

YPS/COBOL

A decorative horizontal band with a red-to-dark-red gradient. It features abstract, glowing white and red lines that swirl and curve across the band, creating a sense of motion and energy.

YPS/COBOL言語文法書

2017年8月版

OSIV MSP/OSIV XSP/Windows

B1FW-5990-02Z0(00)
2017年8月

まえがき

商標について

- ・ Microsoft、Windows、Windows VistaおよびWindows Serverは、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

製品の呼び名について

本書では、各製品を以下のように略記しています。あらかじめご了承ください。

正式名称	略記
Microsoft(R) Windows Server(R) 2016 Datacenter Microsoft(R) Windows Server(R) 2016 Standard Microsoft(R) Windows Server(R) 2016 Essentials	Windows Server 2016
Microsoft(R) Windows Server(R) 2012 R2 Datacenter Microsoft(R) Windows Server(R) 2012 R2 Standard Microsoft(R) Windows Server(R) 2012 R2 Essentials Microsoft(R) Windows Server(R) 2012 R2 Foundation	Windows Server 2012 R2
Microsoft(R) Windows Server(R) 2012 Datacenter Microsoft(R) Windows Server(R) 2012 Standard Microsoft(R) Windows Server(R) 2012 Essentials Microsoft(R) Windows Server(R) 2012 Foundation	Windows Server 2012
Microsoft(R) Windows Server(R) 2008 R2 Datacenter Microsoft(R) Windows Server(R) 2008 R2 Enterprise Microsoft(R) Windows Server(R) 2008 R2 Standard Microsoft(R) Windows Server(R) 2008 R2 Foundation	Windows Server 2008 R2
Microsoft(R) Windows Server(R) 2008 Foundation Microsoft(R) Windows Server(R) 2008 Standard Microsoft(R) Windows Server(R) 2008 Standard without Hyper-V(TM) Microsoft(R) Windows Server(R) 2008 Enterprise Microsoft(R) Windows Server(R) 2008 Enterprise without Hyper-V(TM) Microsoft(R) Windows Server(R) 2008 Datacenter Microsoft(R) Windows Server(R) 2008 Datacenter without Hyper-V(TM)	Windows Server 2008
Windows(R) 10 Home Windows(R) 10 Pro Windows(R) 10 Enterprise Windows(R) 10 Education	Windows 10
Windows(R) 8.1 Windows(R) 8.1 Pro Windows(R) 8.1 Enterprise	Windows 8.1
Windows(R) 7 Home Premium Windows(R) 7 Professional	Windows 7

正式名称	略記
Windows(R) 7 Enterprise Windows(R) 7 Ultimate	
Windows Vista(R) Home Basic Windows Vista(R) Home Premium Windows Vista(R) Business Windows Vista(R) Enterprise Windows Vista(R) Ultimate	Windows Vista

次の製品をすべて指す場合、「Windows」と表記しています。

- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2
- Windows Server 2008
- Windows 10
- Windows 8.1
- Windows 7
- Windows Vista

本書の目的

本書は、富士通 GS/MシリーズおよびWindowsのYPS/COBOLの機能および文法について説明しています。本書で説明する文法は、富士通システム統合アーキテクチャSIA(Systems Integration Architecture)のYPS/COBOL言語共通文法を示しています。本書で説明する文法の範囲でYPS/COBOL言語を使用してプログラムを作成することにより、システム間でのYPS仕様書の再利用性や流通性が保証され、ソフトウェア開発の生産性を向上させることができます。

本書を適用するYPS処理系を、以下に示します。

YPS/COBOL処理系	システム名
MSP YPS/COBOL V12L10 MSP YPS/BASE V12L10	MSP
XSP YPS/COBOL V12L10 XSP YPS/BASE V12L10	XSP
YPS エディタ V5.0、V5.1 YPS WorkBench V5.0、V5.1 YPS/COBOL Std V5.0、V5.1 YPS/COBOL Pro V5.0、V5.1、V5.2、V5.3	Windows

本書の対象読者

本書は、YPS/COBOL言語およびCOBOL言語の基本知識を習得された利用者の方を対象に書かれています。

本書の位置付け

YPS/COBOLの利用方法については、本書のほか以下のマニュアルがあります。

- COBOL言語の文法に関するマニュアル
 - OSIV COBOL文法書
 - NetCOBOL COBOL文法書本書では、上記のマニュアルを総称して、「FUJITSU COBOL文法書」と表記しています。
- YPS仕様書の翻訳から実行までの方法に関するマニュアル
 - OSIV YPS/COBOLコンパイラ使用手引書
 - OSIV YPS支援ツール使用手引書
 - YPS/COBOL V5.1 使用手引書
- COBOL原始プログラムの翻訳から実行までの方法に関するマニュアル
 - OSIV COBOL85使用手引書 基本編
 - OSIV COBOL85使用手引書 インタラクティブデバッグ編
 - OSIV COBOL85メッセージ説明書
 - NetCOBOL ユーザーズガイド

本書の構成

本書は、以下の章で構成されています。

第1章 YPS/COBOL言語の概要

YPS/COBOL言語の概要およびYPS/COBOL言語の構成について説明しています。

第2章 YPS/COBOL言語の基本

YPS仕様書の種類、YPS仕様書の構成、YPS仕様書の構成要素、YPSの機能の概要およびYPSの機能の作用順序について説明しています。

第3章 各表記の仕様

YPS/COBOL仕様書を構成する各表記について、機能、記述形式、記述規則および一般規則を規定しています。また、記述例も記載してあります。

第4章 条件付き翻訳機能

YPS/COBOL言語では、翻訳時に翻訳の対象にする処理および宣言を選択することができます。これを実現する条件付き翻訳機能について規定しています。

第5章 取込み機能

YPS/COBOL言語では、翻訳時に他の仕様書から表記を取り込むことができます。これを実現する取込み機能について規定しています。

第6章 コメントキーワード定義機能

YPS/COBOL言語では、処理や宣言と補足説明(コメント)を、一文で表現することができます。これを実現するコメントキーワード定義機能について規定しています。

第7章 構文定義機能

YPS/COBOL言語では、一連の処理や条件を、利用者が定義する抽象的な表現で記述することができます。これを実現する構文定義機能について規定しています。

第8章 日本語名標宣言機能

YPS/COBOL言語では、COBOL言語の名標、定数および予約語を日本語で記述することができます。これを実現する日本語名標宣言機能について規定しています。

第9章 全角文字記述機能

YPS/COBOL言語では、半角文字で記述しなければならないCOBOL言語の文字列を、全角文字で記述することができます。これを実現する全角文字記述機能について規定しています。

第10章 段階的詳細化

YPS/COBOL言語で、節および節呼出し表記を用いて、トップダウンプログラミングを行う方法について説明しています。

付録A COBOL言語とYPS/COBOL言語の対応

COBOL言語と対比させながら、YPS/COBOL言語を理解したい利用者のために、COBOL言語とYPS/COBOL言語との対応関係を説明しています。

付録B 標準構文

システムが提供している標準構文を示しています。

付録C 標準日本語名標

システムが提供している標準日本語名標を示しています。

付録D 日本語図記号と英語図記号の対応表

YPS/COBOL言語で使用する図記号について、日本語図記号と英語図記号の対応を示しています。

付録E YPS仕様書のファイル形式とコード系

システムごとに規定されている、YPS仕様書のファイル形式と使用されている文字コード系を示しています。

付録F「FUJITSU YPS/COBOL言語文法書 1991年10月版」との相違

「FUJITSU YPS/COBOL言語文法書 1991年10月版」との非互換を列挙しています。

付録G 翻訳指示行に記述する旧仕様に関する指示

翻訳指示行に記述する互換のための指示を示しています。

用語集

本書で使用する用語を一覧し、用語の意味を解説しています。

本書の読み方

“第1章 YPS/COBOL言語の概要”と“第2章 YPS/COBOL言語の基本”は、必ずお読みください。

“第3章 各表記の仕様”から“第10章 段階的詳細化”は、必要に応じてお読みください。この場合、見出し目次を利用することもできます。

COBOL言語に対比させながら、YPS/COBOL言語を理解されたい方は、“付録A COBOL言語とYPS/COBOL言語の対応”をお読みください。

システム固有機能の表示

本書で記述するSIA YPS/COBOL言語共通文法、および本書に記述されるCOBOL言語文法の中には、システムの機能に依存して、仕様が一部異なる部分があります。それらのシステム固有の機能は強調し、強調してある部分の上または左右に、その機能が使用できるシステムを記号で表示しています。記号は、個々のシステム名の総称または略称で表現しています。以下にシステム固有機能の記号を示します。

YPS/COBOL処理系	システム名	総称／略称
MSP YPS/COBOL V12L10 MSP YPS/BASE V12L10	MSP	GS
XSP YPS/COBOL V12L10 XSP YPS/BASE V12L10	XSP	
YPS エディタ V5.0、V5.1 YPS WorkBench V5.0、V5.1 YPS/COBOL Std V5.0、V5.1 YPS/COBOL Pro V5.0、V5.1、V5.2、V5.3	Windows	Win

以下に、システム固有機能の表示例を示します。

- 文中の場合

プログラム内データ部には、基底変数宣言節、ファイル形式宣言節、作業用データ宣言節、定数宣言節、仮引数宣言節、報告書式宣言節[Win]および画面宣言節を構成で示した順序で記述します。

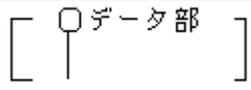
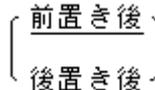
- 図表中の場合

REPORT SECTION.	▽報告書式宣言開始 △報告書式宣言終了
[Win] SCREEN SECTION.	▽画面宣言開始 △画面宣言終了
PROCEDURE DIVISION.	□手続き部

強調部分の機能を使用してプログラムを作成する場合、どのシステムで翻訳/実行するのかについて十分留意してください。システム間での流通性の高いプログラムを作成する場合には、強調部分の機能はなるべく使用しないことをお勧めします。

記述上の約束

本書でYPS仕様書の構成および記述形式の説明に使用している表記法を、以下に示します。

表記法	意味	例	例
[] (角括弧)	省略できることを示します。		データ部が省略できることを示しています。
[] (亀甲括弧)	項目の1つを選択することを示します。		「前置き語」または「後置き語」のどちらか一方を記述することを示しています。
{ } (波括弧)	1回以上繰り返すことを示します。		節を1つ以上記述することを示しています。
<u>下線付き文字列</u> (下線)	利用者がそのまま記述しなければならない部分であることを示します。		「*」はそのまま記述することを示しています。
<i>斜体文字列</i>	利用者が置き換えて記述しなければならない部分であることを示します。	【パラメタ名】	「パラメタ名」は利用者が置き換えて記述することを示しています。
太字文字列	YPSの用語であり、かつ、用語が定義された箇所であることを示します。	順次文	「順次文」という用語がYPSの用語であることを示しています。
⋮ (縦三点リーダ)	任意の表記が記述できることを示します。	▽仮引数宣言開始 ⋮ △仮引数宣言終了	仮引数宣言開始と仮引数宣言終了の間に、任意の表記が記述できることを示しています。

COBOLの用語

本書では、YPS/COBOL言語文法を説明するために、COBOLの用語も使用しています。それらのCOBOLの用語は、各項ごとに「COBOLの用語」として明示してあります。「COBOLの用語」で示された用語の意味、記述規則等については、「FUJITSU COBOL文法書」を参照してください。

また、「COBOLの用語」欄の「⇒」は、索引で調べる場合の用語を表しています。

本書の注意事項

本書の情報はプログラミングサービス情報です。YPS/COBOLを利用した応用プログラムを作成するために利用できます。

本書のコメント

本書で規定するYPS/COBOL言語の仕様は、“FUJITSU YPS/COBOL言語文法書 1991年10月版”の仕様に、COBOL言語との整合性が向上するように機能拡張および改善を行ったものです。また、機能拡張および改善にともない多少の非互換が発生しています。ご注意ください。

お願い

本書を無断で他に転載しないようお願いいたします。

本書は予告なしに変更されることがあります。

輸出管理について

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

2017年8月

Copyright 1990-2017 FUJITSU LIMITED

謝辞

COBOLの言語仕様は、データシステムズ言語協議会(the Conference on Data Systems Languages)の作業によって開発された原仕様に基づくものであり、本書で記述される仕様もまたそれに由来する。データシステムズ言語協議会の要求によって、以下の文章を掲げる。

COBOLは産業界の言語であって、いかなる会社、組織、団体等の占有物でもない。COBOLの委員会は、このプログラミング方式および言語の正確さと機能について、いかなる保証を与えるものでもなく、またそれに関連して、いかなる責任を負うものでもない。

次に示す著作権者は、原仕様書の作成に当たってそれぞれの著作物の一部分を利用することを承認した。この承認は原仕様書をほかのCOBOLの仕様書で利用する場合にまで拡張されるものがある。

- FLOW-MATIC(スペリランド社の商標)Programming for the Univac(R)Iand II,Data Automation Systems,スペリランド社1958年,1959年,著作権.
- IBM Commercial Translator,図書番号,F28-8013,IBM社1959年,著作権.
- FACT,図書番号,27A5260-2760,ミネアポリスハニウェル社1960年,著作権.

目次

第1章 YPS/COBOL言語の概要	1
1.1 YPS/COBOL言語とは	1
1.2 YPS/COBOL言語の構成	1
第2章 YPS/COBOL言語の基本	3
2.1 YPS仕様書で使用できる文字	3
2.2 YPS仕様書	3
2.2.1 YPS仕様書の種類	5
2.2.2 YPS仕様書の構成	6
2.3 YPS表記	7
2.4 制御線	9
2.5 図記号の種類	10
2.6 本文	15
2.6.1 YPSの文字列	16
2.6.2 COBOLの文字列	19
2.6.3 本文の継続方法	20
2.7 YPSの機能	20
2.7.1 YPSの機能の概要	20
2.7.2 YPSの機能の作用順序と作用対象	21
第3章 各表記の仕様	23
3.1 YPS仕様書を構成する表記	23
3.1.1 モジュール関連情報	23
3.1.2 モジュール本体	23
3.2 モジュール関連情報を構成する表記	24
3.2.1 モジュール属性記述行	24
3.2.2 著作権表示行	25
3.2.3 翻訳指示行	25
3.2.4 モジュール概要記述	27
3.3 モジュール本体を構成する表記	27
3.3.1 構文定義部	27
3.3.2 モジュール共通データ部	28
3.3.3 プログラム単位	28
3.4 プログラム単位を構成する表記	30
3.4.1 環境部	31
3.4.2 プログラム内データ部	32
3.4.3 手続き部	34
3.4.4 内部プログラム単位	35
3.5 環境部を構成する表記	37
3.5.1 システム環境宣言節	37
3.5.2 入出力環境宣言節	38
3.6 環境部の節を構成する表記	39
3.7 プログラム内データ部を構成する表記	39
3.7.1 基底変数宣言節	39
3.7.2 ファイル形式宣言節	40
3.7.3 作業用データ宣言節	41
3.7.4 定数宣言節	42
3.7.5 仮引数宣言節	43
3.7.6 報告書式宣言節	44
3.7.7 画面宣言節 [Win]	45
3.8 データ部の節を構成する表記	46
3.8.1 宣言文	47
3.8.2 構造体	48
3.9 手続き部を構成する表記	49
3.9.1 宣言部分	49
3.9.2 節	50

3.10 手続き部の節を構成する表記.....	51
3.10.1 順次系の表記.....	51
3.10.1.1 順次文.....	51
3.10.1.2 空文.....	52
3.10.2 選択系の表記.....	52
3.10.2.1 判定.....	53
3.10.2.2 単一選択(定数指定).....	54
3.10.2.3 単一選択(条件式指定).....	55
3.10.2.4 複合選択(定数指定).....	57
3.10.2.5 複合選択(条件式指定).....	58
3.10.2.6 評価.....	60
3.10.3 ループ系の表記.....	62
3.10.3.1 前判定ループ.....	62
3.10.3.2 後判定ループ.....	63
3.10.3.3 終了条件前判定ループ.....	64
3.10.3.4 変数型ループ.....	65
3.10.3.5 無限ループ.....	68
3.10.3.6 指定回数ループ.....	69
3.10.3.7 更新型ループ.....	70
3.10.3.8 変数型ループと前判定ループの組合せ.....	72
3.10.3.9 変数型ループと後判定ループの組合せ.....	75
3.10.3.10 変数型ループと終了条件前判定ループの組合せ.....	78
3.10.3.11 更新型ループと前判定ループの組合せ.....	81
3.10.3.12 更新型ループと後判定ループの組合せ.....	82
3.10.4 例外処理.....	84
3.10.5 検索処理.....	86
3.10.6 制御変更系の表記.....	88
3.10.6.1 プログラム呼出し.....	88
3.10.6.2 内部プログラム呼出し.....	89
3.10.6.3 二次入口.....	90
3.10.6.4 ラベル.....	91
3.10.6.5 分岐.....	92
3.10.6.6 途中脱出.....	93
3.10.6.7 節呼出し.....	94
3.10.6.8 節出口.....	95
3.10.6.9 復帰.....	95
3.10.6.10 プログラム実行終了.....	96
3.11 内部プログラム単位を構成する表記.....	97
3.12 部および節に依存しない表記.....	97
3.12.1 注釈.....	97
3.12.2 要約定義.....	98
3.12.3 コメント文.....	99
3.12.4 目的言語記述.....	100
3.12.5 強制改ページ.....	101
第4章 条件付き翻訳機能.....	103
4.1 条件付き翻訳機能とは.....	103
4.2 翻訳条件式.....	104
4.3 条件付き翻訳判定.....	105
4.4 条件付き翻訳単一選択(選択条件名指定).....	107
4.5 条件付き翻訳単一選択(翻訳条件式指定).....	108
第5章 取込み機能.....	110
5.1 取込み機能とは.....	110
5.2 取込み.....	111
5.3 置換語句宣言.....	112
5.4 インクルード仕様書.....	115

第6章 コメントキーワード定義機能.....	117
6.1 コメントキーワード定義機能とは.....	117
6.2 コメントキーワード定義文.....	117
6.3 標準コメントキーワード.....	118
6.4 コメントキーワードの引用.....	119
第7章 構文定義機能.....	120
7.1 構文定義機能とは.....	120
7.2 抽象化処理構文定義機能.....	120
7.2.1 抽象化処理構文定義.....	121
7.2.2 抽象化処理構文.....	122
7.2.3 一時変数宣言文.....	122
7.2.4 抽象化処理構文の引用.....	123
7.3 抽象化条件式定義機能.....	127
7.3.1 抽象化条件式定義文.....	127
7.3.2 抽象化条件式構文.....	128
7.3.3 条件式.....	128
7.3.4 抽象化条件式構文の引用.....	130
7.4 構文定義のエスケープ文字.....	130
7.5 外部構文定義仕様書.....	131
第8章 日本語名標宣言機能.....	133
8.1 日本語名標宣言機能とは.....	133
8.2 日本語名標宣言文.....	133
8.3 日本語記述のエスケープ文字.....	135
8.4 外部日本語名標宣言仕様書.....	135
第9章 全角文字記述機能.....	137
9.1 全角文字記述機能とは.....	137
9.2 全角文字記述機能の対象となる文字列.....	137
第10章 段階的詳細化.....	139
10.1 段階的詳細化とは.....	139
付録A COBOL言語とYPS/COBOL言語の対応.....	140
A.1 翻訳単位.....	140
A.2 COBOL原始プログラム.....	140
A.3 見出し部とプログラム終わり見出し.....	141
A.4 環境部.....	141
A.5 データ部.....	142
A.6 手続き部.....	144
A.7 YPS仕様書を記述する上での注意事項.....	150
付録B 標準構文.....	156
B.1 YPS標準構文.....	156
B.2 YPS/COBOL標準構文.....	156
B.3 YPS/COBOL事務処理構文.....	158
付録C 標準日本語名標.....	165
C.1 SIA日本語名標.....	165
C.2 Mシリーズ固有日本語名標 [GS].....	169
付録D 日本語図記号と英語図記号の対応表.....	171
付録E YPS仕様書のファイル形式とコード系.....	175
付録F 「FUJITSU YPS/COBOL言語文法書 1991年10月版」との相違.....	176
付録G 翻訳指示行に記述する旧仕様に関する指示.....	179

用語集.....	180
索引.....	193

第1章 YPS/COBOL言語の概要

本章では、YPS/COBOL言語の概要について説明しています。

1.1 YPS/COBOL言語とは

YPSは、構造化プログラミングの考え方に基づいた表記法を使用し、業務で使用する用語を含む日本語でプログラムを作成できるソフトウェア開発支援システムです。

YPS/COBOL言語は、YPSで事務処理分野のソフトウェアを開発する場合に使用するプログラミング言語です。

YPS/COBOL言語の特長

YPS/COBOL言語には以下に示す特長があります。

仕様書とプログラムの一体化

YPS/COBOL言語文法に基づいて作成したプログラムをYPS仕様書といいます。YPS仕様書はプログラムであると同時に、そのまま仕様書としても使用できます。したがって、開発および保守の段階で、仕様書とプログラムの内容が一致しないというような問題は発生しません。

視覚的な図記号

YPS/COBOL言語では、データの構造や処理の流れを、図記号を使用して表現します。よって、プログラムの内容を視覚的にとらえることができ、プログラムの内容を容易に理解することができます。

日本語による記述

YPS/COBOL言語のプログラムでは、図記号の本文に記述する処理や条件は日本語で記述します。また、COBOL言語の文法がそのまま適用される部分も、YPSの機能(構文定義機能および日本語名標宣言機能)を用いることにより、日本語で記述することができます。

COBOL言語との関係

YPS/COBOL言語はCOBOL言語をベースに設計された言語であり、YPS/COBOL言語の表記は、最終的にはCOBOL言語に対応付けられます。また、YPS/COBOL言語の文法には、COBOL言語の文法がそのまま適用される部分もあります。

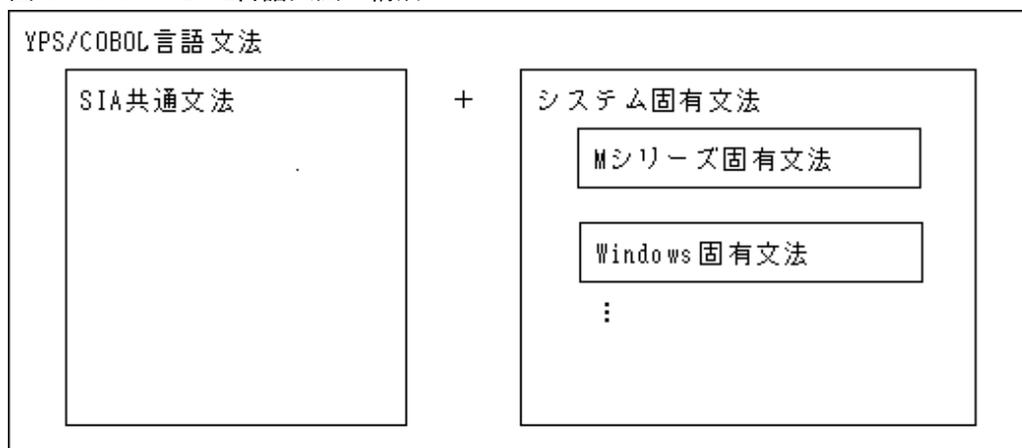
1.2 YPS/COBOL言語の構成

本書で規定しているYPS/COBOL言語文法の構成と各構成要素について説明します。

YPS/COBOL言語文法の構成

YPS/COBOL言語文法は、“[図1.1 YPS/COBOL言語文法の構成](#)”に示すように、SIA YPS/COBOL共通文法(以降、SIA共通文法と略記します)とシステム固有文法から構成されています。

図1.1 YPS/COBOL言語文法の構成



SIA共通文法

SIA共通文法とは、各システムで共通に適用される文法です。

SIA共通文法の範囲で記述したYPS仕様書は^(注)、システム間で流通させることができます。

注: COBOLの文法がそのまま適用される部分は、SIA COBOL共通文法の範囲で記述してください。SIA COBOL共通文法については、“FUJITSU COBOL文法書”を参照してください。

システム固有文法

システム固有文法とは、各システムでのみ適用される文法です。本書では、強調して示してあります。

システム固有文法には、システムに固有なYPS表記とシステムに固有な標準日本語名標とがあります。システムに固有なYPS表記を“表1.1 システムに固有なYPS表記”に示します。システムに固有な日本語名標については、“付録C 標準日本語名標”を参照してください。

表1.1 システムに固有なYPS表記

システムに固有なYPS表記		[GS]	[Win]
データ部	画面宣言節	×	○

○: 指定可能 ×: 指定不可

第2章 YPS/COBOL言語の基本

本章では、YPS/COBOL言語でプログラムを作成するために必要な、YPS/COBOL言語の基本について説明します。

2.1 YPS仕様書で利用できる文字

YPS仕様書で利用できる文字には、半角文字と全角文字とがあります。以下にその分類を示します。また、各システムで利用できるコード系については、“付録E YPS仕様書のファイル形式とコード系”を参照してください。

半角文字:

- 英大文字(A～Zの26文字)
- 数字(0～9の10文字)
- 英小文字(a～zの26文字)^(注1)
- カナ文字(アア～ンの55文字)^(注1)
- 特殊文字^(注2)
- 空白

注1) 英小文字またはカナ文字が使用できないコード系もあります。

注2) コード系により備えている特殊文字に違いはありますが、備えている特殊文字はすべて使用できます。

全角文字:

利用できる文字はコード系により異なります。コード系ごとに、利用できる文字を以下に示します。

- JEF
JIS第一水準文字、JIS第二水準文字、基本非漢字、拡張漢字、拡張非漢字、空白
- シフトJIS
JIS第一水準文字、JIS第二水準文字、JIS非漢字
- EUC
JIS第一水準文字、JIS第二水準文字、JIS非漢字、JEF拡張漢字、JEF拡張非漢字

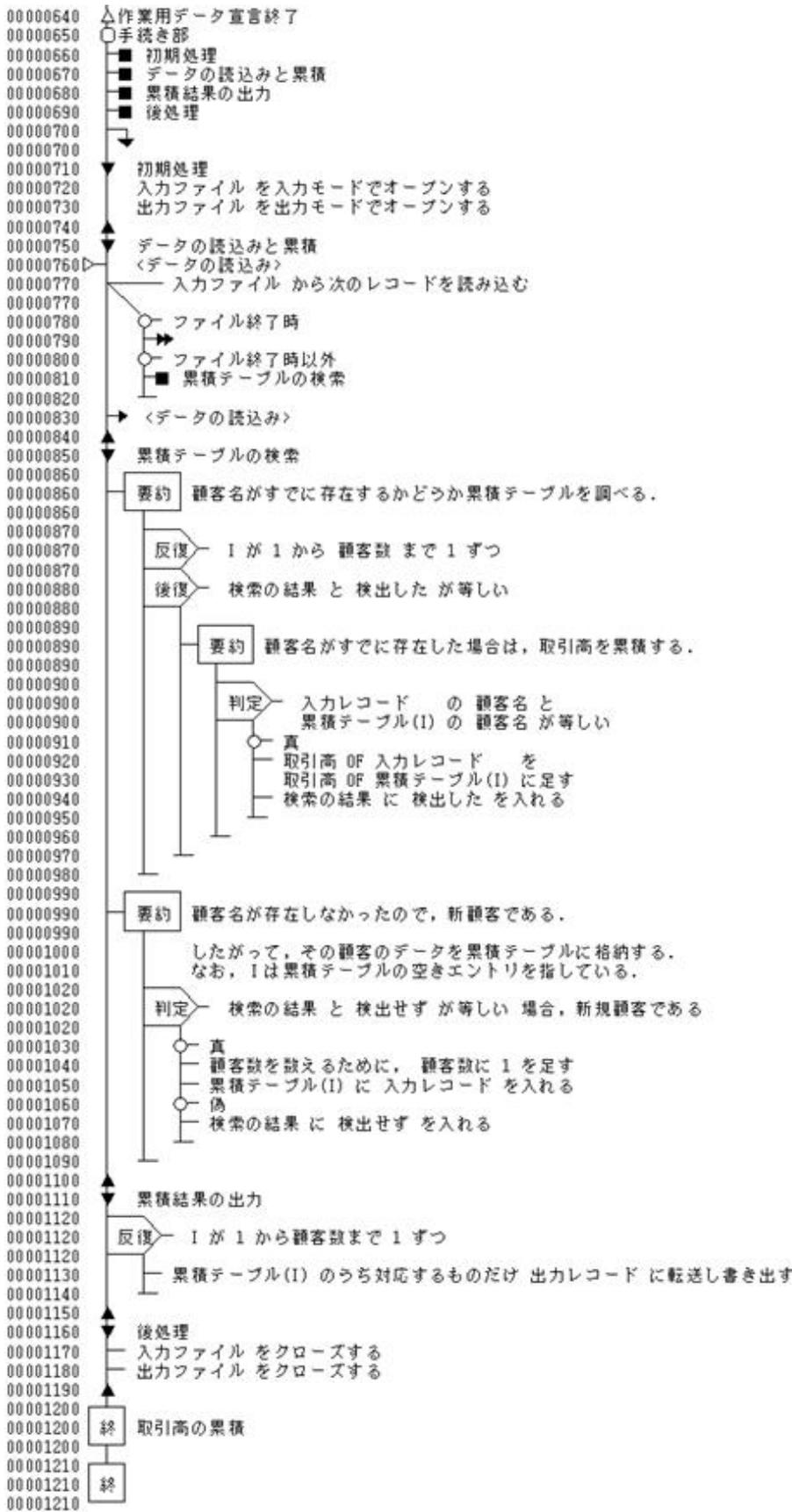
2.2 YPS仕様書

YPS/COBOL言語の文法に基づいて記述されたプログラムのことを**YPS仕様書**といいます。また、YPS仕様書のことを**モジュール**ということもあります。

YPS仕様書の記述例を“[図2.1 YPS仕様書の記述例](#)”に示します。

図2.1 YPS仕様書の記述例





2.2.1 YPS仕様書の種類

YPS仕様書には、以下の4種類があります。目的により、使い分けてください。

- ・ プログラム仕様書
- ・ インクルード仕様書
- ・ 外部構文定義仕様書
- ・ 外部日本語名標宣言仕様書

プログラム仕様書

プログラム仕様書は、翻訳時に主な入力となるYPS仕様書です。“[図2.1 YPS仕様書の記述例](#)”に示したYPS仕様書は、プログラム仕様書です。

インクルード仕様書

インクルード仕様書は、プログラムの一部が記述されたYPS仕様書です。

YPSの取込み機能により、インクルード仕様書の内容を、他のYPS仕様書で取り込むことができます。

取込み機能については、“[第5章 取込み機能](#)”を参照してください。

外部構文定義仕様書

外部構文定義仕様書は、複数のYPS仕様書で引用するコメントキーワード、抽象化処理構文および抽象化条件式構文を定義するYPS仕様書です。

プログラム仕様書の翻訳時に外部構文定義仕様書を指定することにより、当仕様書で定義したコメントキーワード、抽象化処理構文および抽象化条件式構文が引用できるようになります。

コメントキーワードについては、“[第6章 コメントキーワード定義機能](#)”を、抽象化処理構文および抽象化条件式構文については、“[第7章 構文定義機能](#)”を参照してください。

外部日本語名標宣言仕様書

外部日本語名標宣言仕様書は、複数のYPS仕様書で引用する日本語名標を宣言するYPS仕様書です。

プログラム仕様書の翻訳時に外部日本語名標宣言仕様書を指定することにより、当仕様書で宣言した日本語名標が引用できるようになります。

日本語名標については、“[第8章 日本語名標宣言機能](#)”を参照してください。

2.2.2 YPS仕様書の構成

YPS仕様書は、仕様書の種類に関係なく、次の2つの表記から構成されます。“[図2.2 YPS仕様書の構成](#)”を参照してください。

モジュール関連情報

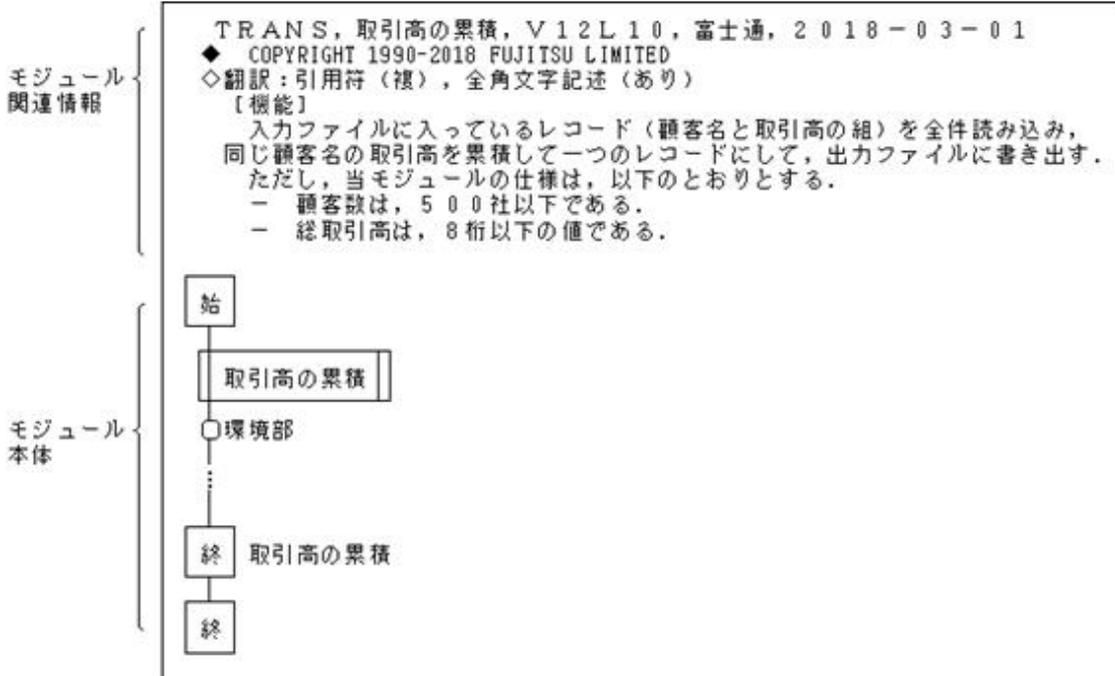
YPS仕様書の機能概要やモジュール名など、モジュールに関する情報を記述するところです。

モジュール本体

データや環境の宣言および処理手続きなど、モジュールの実体を記述するところです。

モジュール関連情報およびモジュール本体の詳細については、“[第3章 各表記の仕様](#)”を参照してください。

図2.2 YPS仕様書の構成



2.3 YPS表記

YPS表記とは、YPS仕様書を構成する単位です。単に表記ということもあります。

先に説明したモジュール関連情報およびモジュール本体もYPS表記ですし、モジュール関連情報およびモジュール本体を構成する要素もYPS表記です。

モジュール関連情報の表記

モジュール関連情報は、以下の4つの表記で構成します。

- ・ モジュール属性記述行
- ・ 著作権表示行
- ・ 翻訳指示行
- ・ モジュール概要記述

モジュール関連情報の記述例を“[図2.3 モジュール関連情報の表記](#)”に示します。

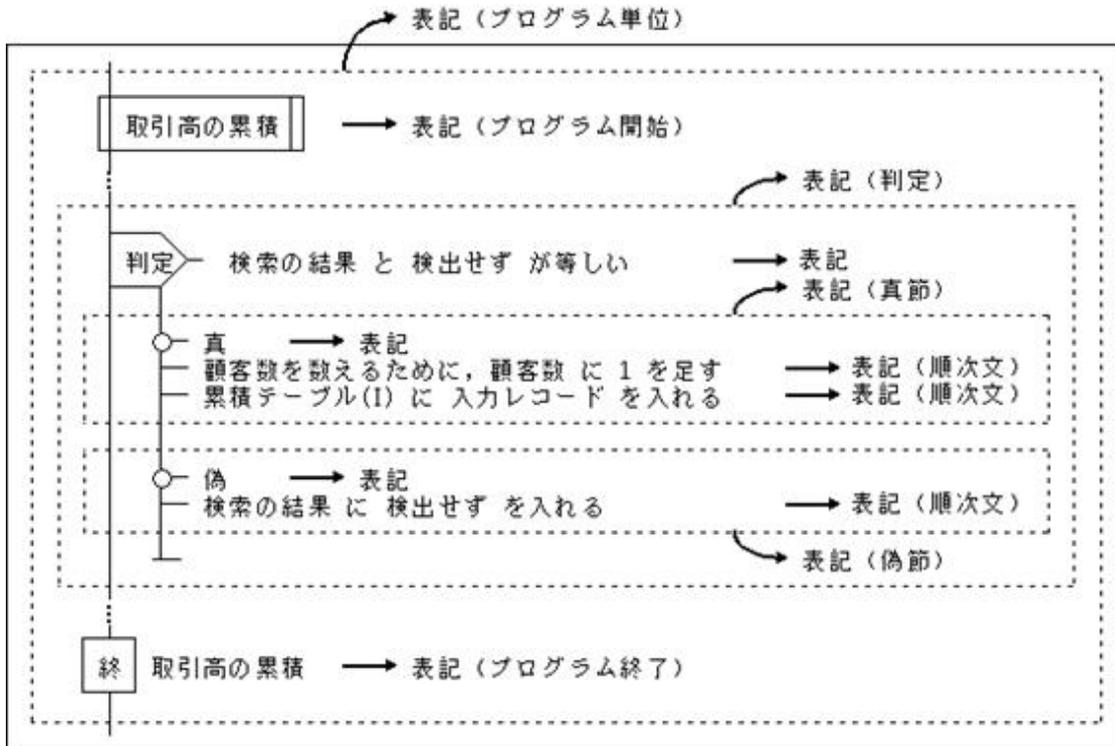
図2.3 モジュール関連情報の表記



モジュール本体の表記

モジュール本体において、図記号と本文の組合せであれば、どんなに小さな単位でもYPS表記です。例を“[図2.4 モジュール本体の表記](#)”に示します。なお、括弧内の名称はYPS表記名を示しています。

図2.4 モジュール本体の表記



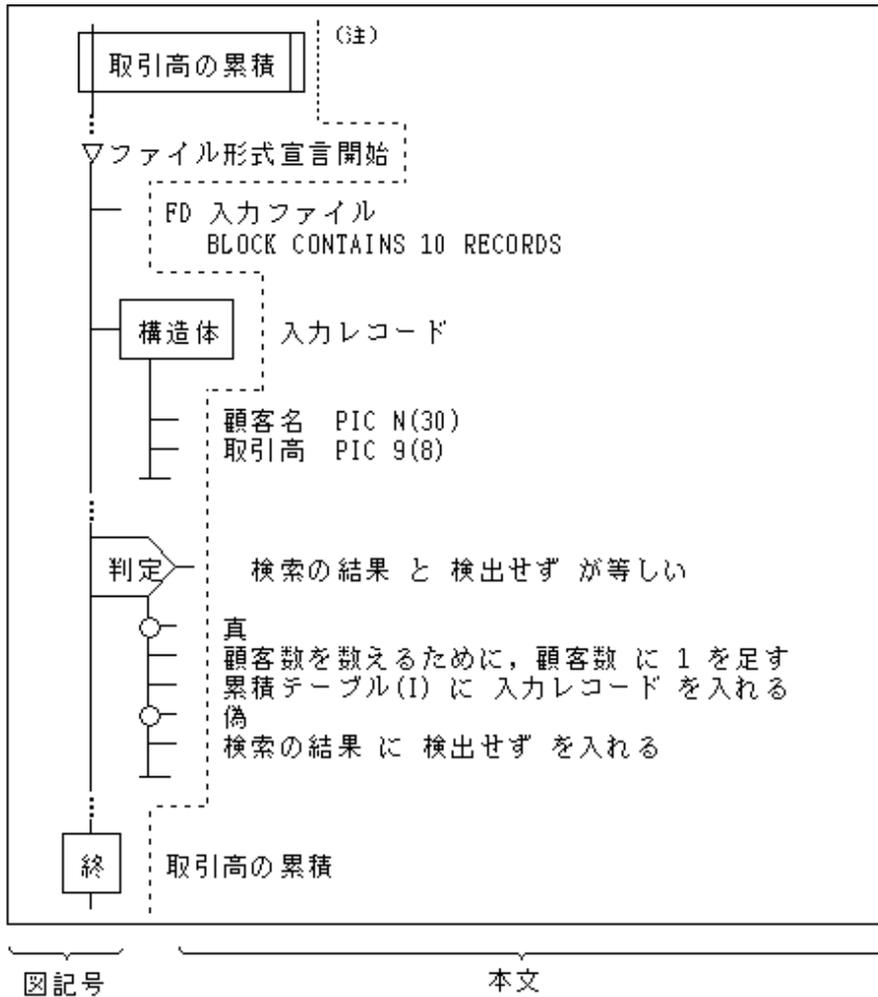
YPS表記は、図記号と本文から構成されます。

図記号とは、YPS表記編集でYPS仕様書を作成する時に、メニューから選択する部分です。プログラムの制御構造を表現しています。

本文とは、YPS表記編集でYPS仕様書を作成する時に、図記号の右側または中に利用者が直接記述する部分です。

図記号と本文の例を“[図2.5 図記号と本文](#)”に示します。

図2.5 図記号と本文



(注) プログラム名「取引高の累積」は本文です。

2.4 制御線

制御線とは、モジュール本体において、図記号と図記号を繋ぐ縦の線のことです。

制御線には、物理的な制御線と論理的な制御線があります。

物理的な制御線

物理的な制御線とは、YPS仕様書に記述された、見た目そのものの制御線のことです。

論理的な制御線

論理的な制御線とは、以下に示すYPSの機能およびYPS表記の解釈の結果、生成されるYPS仕様書の制御線のことです。

- YPSの機能
 - 条件付き翻訳機能による翻訳対象の取捨選択
 - 取込み機能によるインクルード仕様書の取込み
 - 抽象化処理構文定義機能における引用文の展開
- YPSの表記
 - 要約定義表記の解釈

主制御線

制御線のうち、最も左端にある制御線を**主制御線**といいます。主制御線にも、物理的な主制御線と論理的な主制御線があります。本書で主制御線と記述している場合には、特に断りのない限り、論理的な主制御線の意味で使用しています。

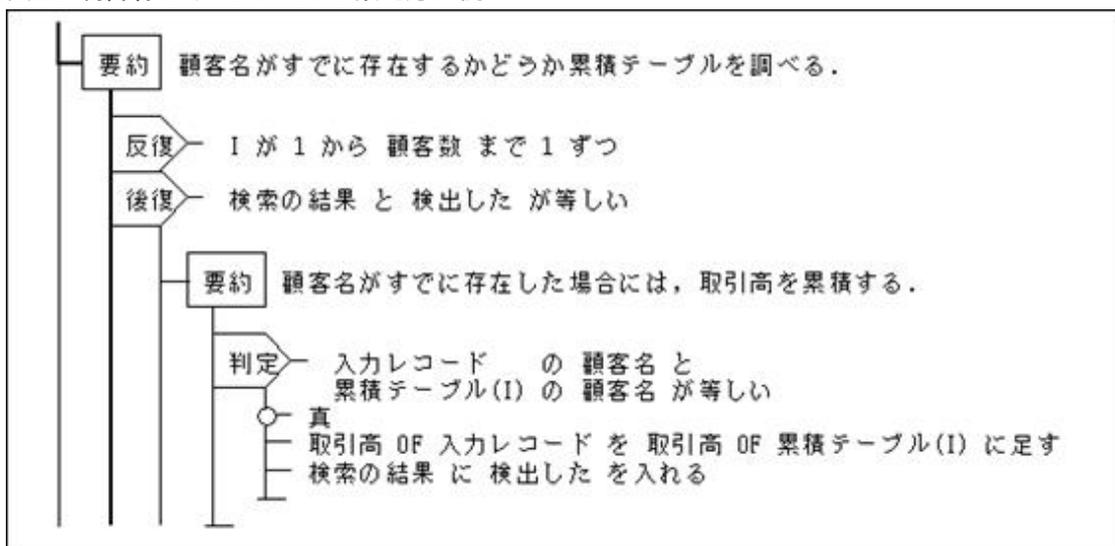
制御線のネスト

ある表記を別の表記の中に記述すると、制御線が多重になることがあります。このように制御線が多重になることを**制御線のネスト**といいます。また、その多重度を**制御線のネストレベル**といいます。

制御線のネストレベルは、主制御線を0とし、右にいくほど数字が大きくなります。YPS仕様書に記述できる物理的な制御線のネストレベルは10までです。

制御線のネストレベルの数え方の例を“[図2.6 制御線のネストレベルの数え方の例](#)”に示します。

図2.6 制御線のネストレベルの数え方の例



このYPS仕様書において、太線で示している制御線が論理的な主制御線です。また、判定表記が記述されている制御線のネストレベルは以下のとおりです。

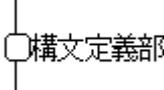
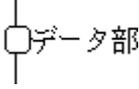
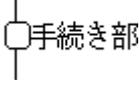
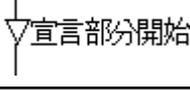
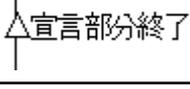
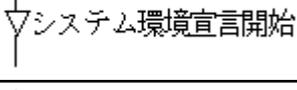
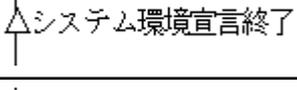
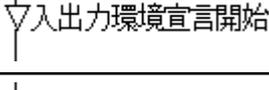
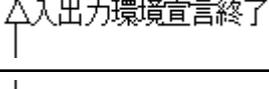
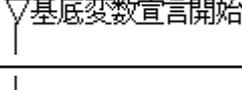
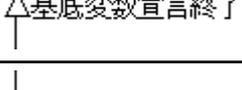
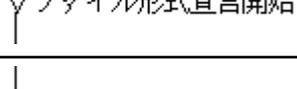
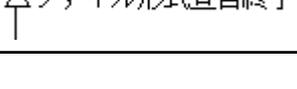
- ・ 物理的な制御線のネストレベル: 3
- ・ 論理的な制御線のネストレベル: 1

2.5 図記号の種類

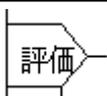
図記号名、図記号および図記号を利用するYPS表記の一覧を“[表2.1 図記号の一覧](#)”に示します。

表2.1 図記号の一覧

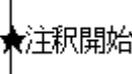
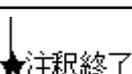
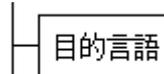
種別	図記号名	図記号	図記号を利用するYPS表記
モジュール本体	モジュール本体開始図記号	始	3.1.2 モジュール本体
終了	終了図記号	終 または 終	3.1.2 モジュール本体 3.3.3 プログラム単位 3.4.4 内部プログラム単位 7.2.1 抽象化処理構文定義
プログラム単位	プログラム開始図記号		3.3.3 プログラム単位

種別	図記号名	図記号	図記号を利用するYPS表記	
内部プログラム単位	内部プログラム開始図記号		3.4.4 内部プログラム単位	
部	構文定義部開始図記号		3.3.1 構文定義部	
	環境部開始図記号		3.4.1 環境部	
	データ部開始図記号		3.3.2 モジュール共通データ部 3.4.2 プログラム内データ部	
	手続き部開始図記号		3.4.3 手続き部	
宣言部分	宣言部分開始図記号		3.9.1 宣言部分	
	宣言部分終了図記号			
節	環境部の節	システム環境宣言開始図記号		3.5.1 システム環境宣言節
		システム環境宣言終了図記号		
		入出力環境宣言開始図記号		3.5.2 入出力環境宣言節
		入出力環境宣言終了図記号		
	データ部の節	基底変数宣言開始図記号		3.7.1 基底変数宣言節
		基底変数宣言終了図記号		
		ファイル形式宣言開始図記号		3.7.2 ファイル形式宣言節
		ファイル形式宣言終了図記号		

種別	図記号名	図記号	図記号を利用するYPS表記
	作業用データ宣言開始 図記号	 作業用データ宣言開始	3.7.3 作業用データ宣言節
	作業用データ宣言終了 図記号	 作業用データ宣言終了	
	定数宣言開始図記号	 定数宣言開始	3.7.4 定数宣言節
	定数宣言終了図記号	 定数宣言終了	
	仮引数宣言開始図記号	 仮引数宣言開始	3.7.5 仮引数宣言節
	仮引数宣言終了図記号	 仮引数宣言終了	
	報告書式宣言開始図記号	 報告書式宣言開始	3.7.6 報告書式宣言節
	報告書式宣言終了図記号	 報告書式宣言終了	
	[Win] 画面宣言開始図記号	 画面宣言開始	3.7.7 画面宣言節 [Win]
	[Win] 画面宣言終了図記号	 画面宣言終了	
手続き部の節	節開始図記号(詳細部開始 図記号)		3.9.2 節(詳細部)
	節終了図記号(詳細部終了 図記号)		
構造体	構造体図記号		3.8.2 構造体
順次 宣言 定義	順次図記号		3.8.1 宣言文 3.10.1.1 順次文 3.10.1.2 空文 3.12.3 コメント文 6.2 コメントキーワード定義文 7.2.3 一時変数宣言文

種別	図記号名	図記号	図記号を利用するYPS表記
			7.2.4 抽象化処理構文の引用 7.3.1 抽象化条件式定義文 8.2 日本語名標宣言文
選択	判定図記号		3.10.2.1 判定
	選択図記号		3.10.2.2 単一選択(定数指定) 3.10.2.3 単一選択(条件式指定)
	複選図記号		3.10.2.4 複合選択(定数指定) 3.10.2.5 複合選択(条件式指定)
	評価図記号		3.10.2.6 評価
ループ	前復図記号		3.10.3.1 前判定ループ
	後復図記号		3.10.3.2 後判定ループ
	終復図記号		3.10.3.3 終了条件前判定ループ
	反復図記号		3.10.3.4 変数型ループ 3.10.3.5 無限ループ 3.10.3.6 指定回数ループ
	更復図記号		3.10.3.7 更新型ループ
ループの組合せ	反復と前復の組合せ図記号		3.10.3.8 変数型ループと前判定ループの組合せ
	反復と後復の組合せ図記号		3.10.3.9 変数型ループと後判定ループの組合せ
	反復と終復の組合せ図記号		3.10.3.10 変数型ループと終了条件前判定ループの組合せ

種別	図記号名	図記号	図記号を利用するYPS表記
	更復と前復の組合せ図記号		3.10.3.11 更新型ループと前判定ループの組合せ
	更復と後復の組合せ図記号		3.10.3.12 更新型ループと後判定ループの組合せ
例外処理	例外処理図記号		3.10.4 例外処理 3.10.5 検索処理 7.2.4 抽象化処理構文の引用
選択節	丸図記号		3.10.2.1 判定 3.10.2.2 単一選択(定数指定) 3.10.2.3 単一選択(条件式指定) 3.10.2.4 複合選択(定数指定) 3.10.2.5 複合選択(条件式指定) 3.10.2.6 評価 3.10.4 例外処理 3.10.5 検索処理 4.3 条件付き翻訳判定 4.4 条件付き翻訳単一選択(選択条件名指定) 4.5 条件付き翻訳単一選択(翻訳条件式指定)
制御変更	プログラム呼出し図記号		3.10.6.1 プログラム呼出し
	内部プログラム呼出し図記号		3.10.6.2 内部プログラム呼出し
	二次入口図記号		3.10.6.3 二次入口
	ラベル図記号		3.10.6.4 ラベル
	分岐図記号		3.10.6.5 分岐
	途中脱出図記号		3.10.6.6 途中脱出

種別	図記号名	図記号	図記号を利用するYPS表記
	節呼出し図記号(概要部図記号)		3.10.6.7 節呼出し(概要部)
	節出口図記号		3.10.6.8 節出口
	復帰図記号		3.10.6.9 復帰
	実行終了図記号		3.10.6.10 プログラム実行終了
注釈	注釈開始図記号		3.12.1 注釈
	注釈終了図記号		
要約定義	要約図記号		3.12.2 要約定義
目的言語	目的言語図記号		3.12.4 目的言語記述
改ページ	改ページ図記号		3.12.5 強制改ページ
条件付き翻訳	条件付き翻訳判定図記号		4.3 条件付き翻訳判定
	条件付き翻訳選択図記号		4.4 条件付き翻訳単一選択(選択条件名指定) 4.5 条件付き翻訳単一選択(翻訳条件式指定)
取込み	取込み図記号		5.2 取込み
構文定義	抽象化処理構文定義開始図記号		7.2.1 抽象化処理構文定義

2.6 本文

本文とは、利用者が各YPS表記の記述規則に従って記述する、ひとつながりの文字の列です。本節では、本文を構成する要素であるYPSの文字列およびCOBOLの文字列と、本文の継続方法について説明します。

2.6.1 YPSの文字列

YPSの文字列は、YPSの文法により規定される文字列です。

YPSの文字列には、以下に示すものがあります。

- YPSのキーワード
- YPSの識別子
- YPSの演算子
- YPSのコメント
- YPSの区切り文字

YPSのキーワード

YPSのキーワードは、YPSの文法で規定される特定の文字列です。

YPSのキーワードは、システム定義のキーワードと利用者定義のキーワードに分類されます。

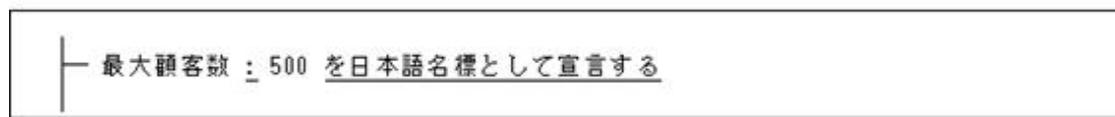
システム定義のキーワード

システム定義のキーワードは、YPS表記の記述形式として規定されている文字列および記号です。コメント記述文字、構文定義のエスケープ文字および日本語記述のエスケープ文字も、システム定義のキーワードです。本書の記述形式では、下線で示してあります。

システム定義のキーワードの例を“[図2.7 システム定義のキーワードの例](#)”に示します。

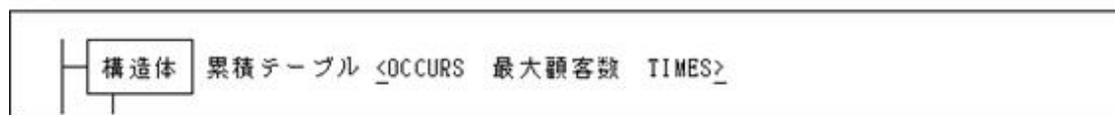
図2.7 システム定義のキーワードの例

【例1】



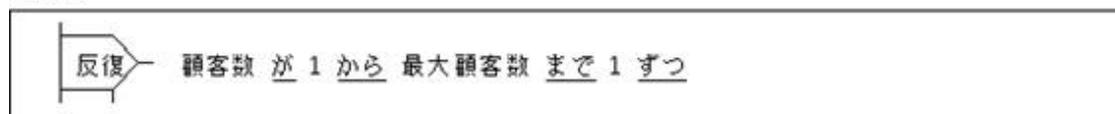
最大顧客数 : 500 を日本語名標として宣言する

【例2】



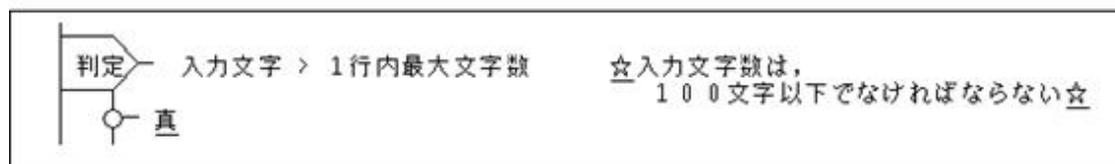
構造体 累積テーブル <OCCURS 最大顧客数 TIMES>

【例3】



反復 顧客数 が 1 から 最大顧客数 まで 1 ずつ

【例4】



判定 入力文字 > 1行内最大文字数 ☆入力文字数は、100文字以下でなければならない☆

備考: 下線部分がシステム定義のキーワードです。

利用者定義のキーワード

利用者定義のキーワードは、構文定義機能およびコメントキーワード定義機能により、キーワードとして定義された文字列です。標準構文のキーワードおよび標準コメントキーワードも利用者定義のキーワードに含まれます。

利用者定義のキーワードの例を“[図2.8 利用者定義のキーワードの例](#)”に示します。

図2.8 利用者定義のキーワードの例



備考: 下線部分が利用者定義のキーワードです。

YPSの識別子

YPSの識別子は、YPSの機能を構成する要素に対して、利用者が付けた名前です。

YPSの識別子には、以下に示すものがあります。

- ・ 条件付き翻訳の翻訳条件名および選択条件名
- ・ 取込みの取込み名およびライブラリ名
- ・ 置換語句宣言の語1～語4
- ・ 抽象化処理構文定義および抽象化条件式定義文のパラメタ名
- ・ 抽象化処理構文定義の一時変数名
- ・ 日本語名標宣言文の文字列1

YPSの識別子には、その識別子を記述する場所でシステム定義のキーワードとなっている文字列と同じ文字列を含めることはできません。その他の記述規則等については、それぞれの機能の説明を参照してください。

YPSの演算子

YPSの演算子には、日本語論理演算子があります。

日本語論理演算子には、以下に示す二つの演算子があります。

- ・ かつ
- ・ または

日本語論理演算子の詳細については、“[7.3.3 条件式](#)”を参照してください。

YPSのコメント

YPSのコメントは、プログラムの意味に何の影響も与えない文字列です。コメントの文字列には、特に規定のない限り、YPS仕様書で使用できる文字はすべて記述することができます。

YPSのコメントには以下のものがあります。

- ・ モジュール関連情報内
 - モジュール属性記述行内の各項目
 - 著作権表示
 - モジュール概要記述
- ・ モジュール本体内
 - コメント記述文字によるコメント
 - 要約定義によるコメント
 - コメントキーワードによるコメント

モジュール属性記述行内の各項目

モジュール属性記述行内の各項目は、YPS仕様書の見出しを記述するためのコメントです。

著作権表示

著作権表示は、YPS仕様書の著作権の所在を明示するためのコメントです。

モジュール概要記述

モジュール概要記述は、機能やインターフェースなど、仕様書の概要について記述するためのコメントです。

コメント記述文字によるコメント

コメント記述文字(「☆」)によるコメントは、本文中の任意の位置にコメントを記述するためのコメントです。

順次図記号の本文にコメント記述文字によるコメントだけを記述した表記を、コメント文といいます。コメント文の詳細については、“[3.12.3 コメント文](#)”を参照してください。

コメントの記述方法

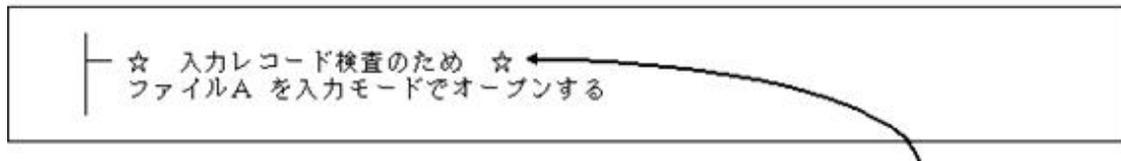
- a. コメントは、モジュール本体内のYPS表記の本文に記述します。ただし、以下の場所では、文字「☆」はコメント記述文字とは扱われないので、コメントを記述することはできません。
 - 文字列リテラル
 - 目的言語記述
- b. コメントは、一つの本文中で、二つのコメント記述文字で囲って記述します。
- c. 一つの本文中に複数のコメントを記述することもできます。その場合、本文の先頭から最初に現れたコメント記述文字がコメントの開始、次に現れたコメント記述文字がコメントの終了とみなされます。本文の残りの部分についても同様です。
- d. 本文がコメントで終わる場合、コメントの終了を表すコメント記述文字は省略することができます。

コメントの記述例

コメント記述文字によるコメントの記述例を、“[図2.9 コメント記述文字によるコメントの記述例](#)”に示します。

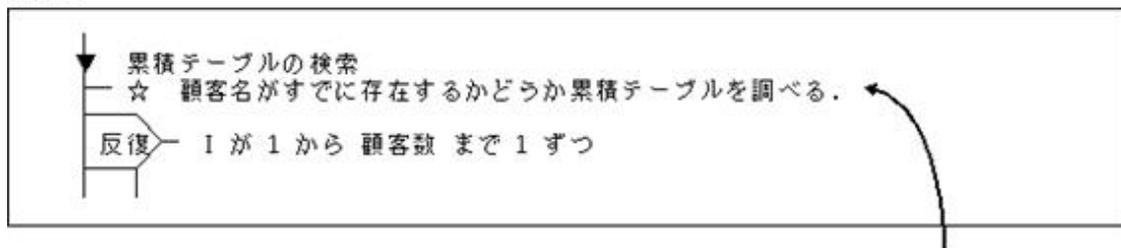
図2.9 コメント記述文字によるコメントの記述例

【例1】



本文の途中であるため、コメント記述文字は省略することができません。

【例2】



本文の終わりであるため、コメント記述文字は省略することができます。

要約定義によるコメント

要約定義によるコメントは、ひとまとまりの処理や宣言に対して、その概要を説明するためのコメントです。

要約定義の詳細については、“[3.12.2 要約定義](#)”を参照してください。

コメントキーワードによるコメント

コメントキーワードによるコメントは、本文に記述された処理や宣言に対して、その目的や理由を補足説明するコメントを付加し、処理や宣言と1文化した本文を記述するためのコメントです。

コメントキーワードについては、“[第6章 コメントキーワード定義機能](#)”を参照してください。

YPSの区切り文字

YPSの区切り文字は、YPSの文法により規定される部分において、本文の要素を区切るための文字です。YPSの区切り文字には、以下の文字があります。

- ・ モジュール属性記述行における全角または半角の「,」または全角の「、」
- ・ 翻訳指示行における全角または半角の「,」
- ・ モジュール本体内における全角または半角の空白列
空白列は特に断りがない限り省略することができます。

2.6.2 COBOLの文字列

COBOLの文字列は、COBOLの文法により規定される文字列です。

COBOLの文字列に対してYPSの文法が影響するのは、以下の項目です。

- ・ 記述文字
- ・ 定数

記述文字

YPSの全角文字記述機能を使用することにより、COBOL言語では半角文字で記述しなければならない文字列(語、数字定数、予約語など)を、全角文字で記述することができます。

全角文字記述機能については、“[第9章 全角文字記述機能](#)”を参照してください。

定数

YPSでは、定数のうち引用符で囲んで値を表すものを総称して、**文字列リテラル**といいます。文字列リテラル内の文字列は、値そのものを表し、YPSの機能の作用対象にはなりません。

文字列リテラルの記述規則

COBOL言語の文字定数の記述規則に加え、以下の規則が追加されます。

- a. 文字列リテラル内の文字列の中に全角の引用符と同じ文字を含めたいときには、全角文字記述機能を使用している場合でも、文字列リテラルの両端の引用符は半角で記述します。

文字列リテラルの記述例

文字列リテラルの記述例を、“[図2.10 文字列リテラルの記述例](#)”に示します。

図2.10 文字列リテラルの記述例

【例1】

文字列リテラル内なので、コメントキーワードとはみなされません。

```
| DISPLAY NC "データ取り出しのため、2分ほどお待ちください。"
```

全角文字記述機能を使用していれば、引用符は全角で記述できます。

【例2】

```
| DISPLAY NC " "データエラーが発生しました。"
```

引用符と同じ文字が含まれているので、引用符は半角で記述しています。

備考:「”」は全角の引用符を表しています。

2.6.3 本文の継続方法

本文を継続するとは、一つの表記の本文を2行以上に渡って記述することをいいます。また、2行目以降を**継続行**といいます。

本文を継続する場合の規則を、以下に説明します。

目的言語記述の本文の継続

目的言語記述の本文を継続する場合には、COBOL言語の文法に従って継続します。

目的言語記述以外の表記の本文の継続

目的言語記述以外の表記の本文を継続する場合には、任意の位置で継続することができます。

2.7 YPSの機能

YPSの機能には、以下に示す機能があります。

- ・ 条件付き翻訳機能
- ・ 取込み機能
- ・ コメントキーワード定義機能
- ・ 構文定義機能
- ・ 日本語名標宣言機能
- ・ 全角文字記述機能

本節では、これらの機能の概要、作用順序および作用対象について説明します。

2.7.1 YPSの機能の概要

それぞれのYPSの機能の概要は以下のとおりです。

条件付き翻訳機能

条件付き翻訳機能とは、翻訳時のオプションにより、翻訳の対象にする処理および宣言を切り分ける機能です。

条件付き翻訳機能を使用することにより、対象システムの違いなどにより処理および宣言を変更する必要があるプログラムの、仕様書の一元管理が行えます。

取込み機能

取込み機能とは、プログラムの一部を他のYPS仕様書から取り込む機能です。また、取込み時に、文字列および文字列の一部を他の文字列に置き換えることもできます。

取込み機能を使用することにより、処理および宣言の部品化が行えます。

コメントキーワード定義機能

コメントキーワード定義機能とは、本文中の処理や宣言と補足説明(コメント)とを1文化して記述するための、コメント付加機能です。

構文定義機能

構文定義機能とは、一連の処理や条件をプロジェクトの用語を使った日本語で記述するための機能です。

日本語名標宣言機能

日本語名標宣言機能とは、本文中に記述するCOBOLの文字列(語、定数、予約語など)に対して、日本語の名標を付けるための、名標置き換え機能です。

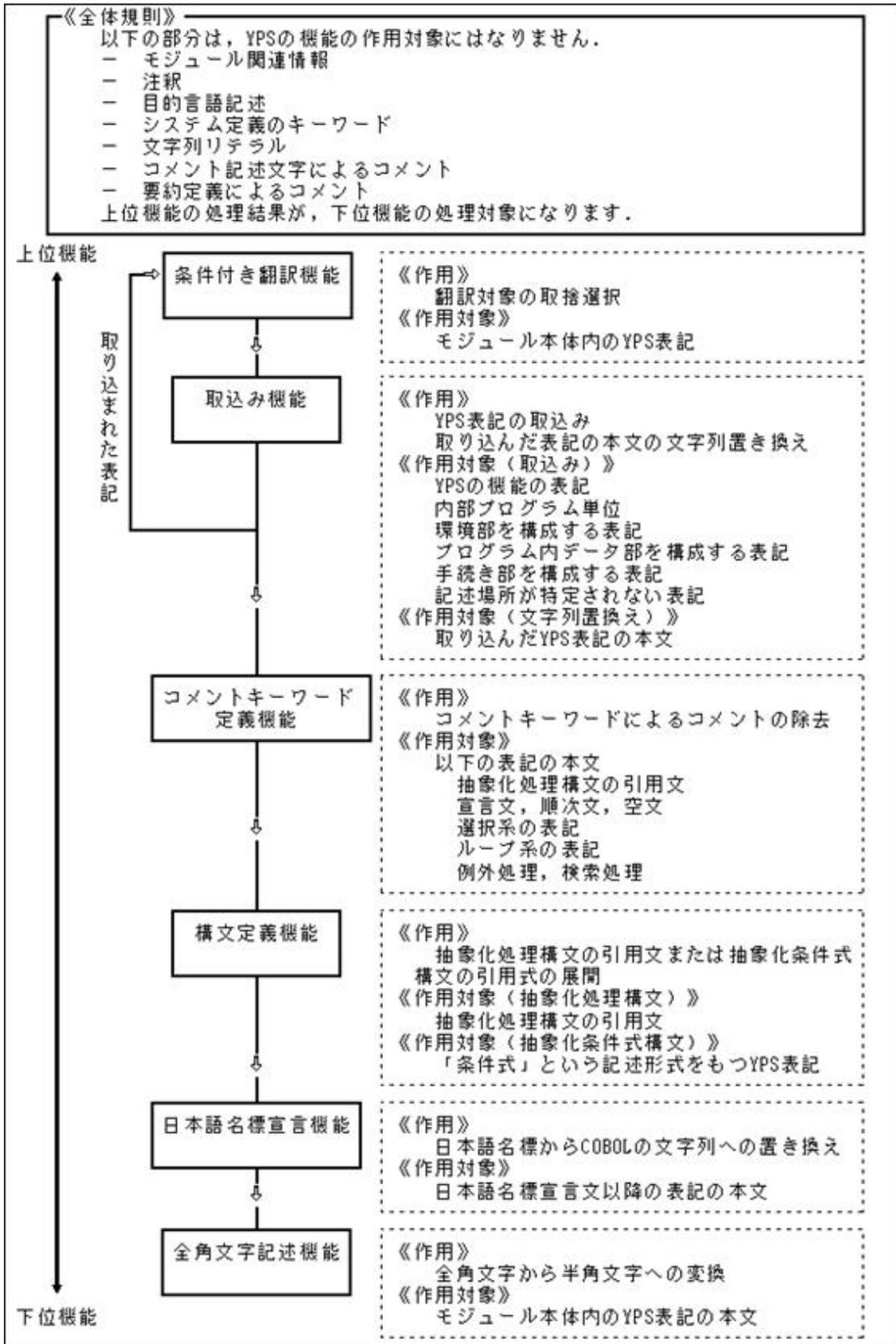
全角文字記述機能

全角文字記述機能とは、COBOLでは半角文字で記述しなければならない文字列を、全角文字で記述できるようにするための、全角文字から半角文字への変換機能です。

2.7.2 YPSの機能の作用順序と作用対象

YPS仕様書の翻訳過程における、YPSの機能の作用順序と作用対象を“[図2.11 YPSの機能の作用順序と作用対象](#)”に示します。

図2.11 YPSの機能の作用順序と作用対象



第3章 各表記の仕様

本章では、YPS仕様書を構成する各表記について説明します。

本章では、YPSの機能を使用した場合の規則については説明していません。YPSの機能を使用する場合には、第4章以降のYPSの機能の説明を参照してください。

3.1 YPS仕様書を構成する表記

YPS仕様書を構成する表記には、以下の2つの表記があります。

- ・ モジュール関連情報
- ・ モジュール本体

3.1.1 モジュール関連情報

モジュール関連情報は、YPS仕様書の機能概要やモジュール名など、モジュールに関する情報を記述するところです。

構成

モジュール属性記述行	→	TRANS, 取引高の累積, V12L10, 富士通, 2018-03-
著作権表示行	→	◆ COPYRIGHT 1990-2018 FUJITSU LIMITED
翻訳指示行	→	◇ 翻訳: 引用符(複), 全角文字記述(あり)
モジュール概要記述	{	[機能] 入力ファイルに入っているレコード(顧客名と取引高の組)を全件 同じ顧客名の取引高を累積して一つのレコードにして, 出力ファイル ただし, 当モジュールの仕様は, 以下のとおりとする. - 顧客数は, 500社以下である. - 総取引高は, 8桁以下の値である.

記述規則

モジュール関連情報には、モジュール属性記述行、著作権表示行、翻訳指示行およびモジュール概要記述を、構成で示した順序で記述します。ただし、翻訳指示行はプログラム仕様書にだけ記述できます。

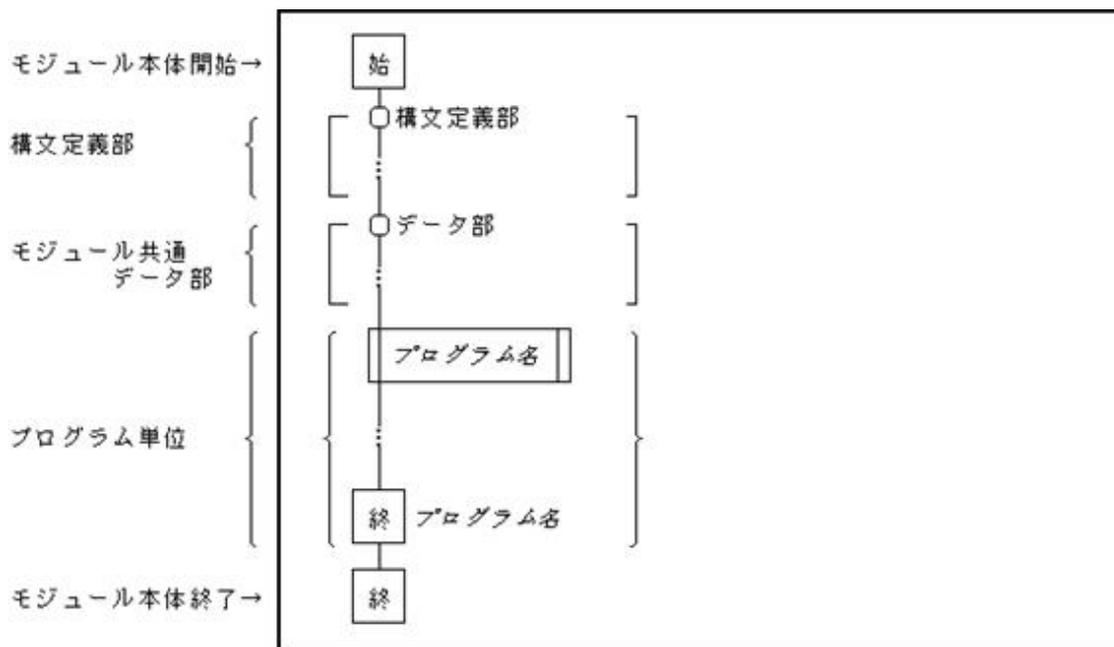
著作権表示行、翻訳指示行およびモジュール概要記述は、省略することができます。

3.1.2 モジュール本体

モジュール本体は、データや環境の宣言、処理手続き、利用者定義構文の定義など、モジュールの実体を記述するところです。

本項では、プログラム仕様書のモジュール本体の記述規則について説明します。

構成



記述規則

プログラム仕様書のモジュール本体には、1つの構文定義部、1つのモジュール共通データ部および1つ以上のプログラム単位を構成で示した順序で記述します。構文定義部およびモジュール共通データ部は、省略することができます。

3.2 モジュール関連情報を構成する表記

モジュール関連情報を構成する表記には、以下の4つの表記があります。

- ・ モジュール属性記述行
- ・ 著作権表示行
- ・ 翻訳指示行
- ・ モジュール概要記述

3.2.1 モジュール属性記述行

モジュール属性記述行は、YPS仕様書の見出しです。

記述形式

[モジュール名] , [機能概要] , [バージョンレベル] , [担当者名] , [作成日付]

記述規則

モジュール属性記述行の本文には、モジュール名、機能概要、バージョンレベル、担当者名および作成日付を記述形式に示した順序で記述します。モジュール属性記述行の各項目は省略することができます。

モジュール属性記述行の各項目は、全角の「,」、半角の「,」または全角の「,」で区切ります。したがって、各項目には、上記の区切り文字を含めることはできません。

モジュール名

モジュール名には、モジュールの名前を8文字以内で記述します。

全角および半角の英大文字、英小文字、数字、特殊文字で記述します。

機能概要

機能概要には、モジュールの機能概要を20文字以内で記述します。

YPS仕様書で使用できる文字で記述します。

バージョンレベル

バージョンレベルには、モジュールのバージョンレベルを6文字以内で記述します。

全角および半角の英大文字、英小文字、数字、特殊文字で記述します。

担当者名

担当者名には、モジュールの作成担当者の名前を6文字以内で記述します。

YPS仕様書で使用できる文字で記述します。

作成日付

作成日付には、モジュールを作成した日付を10文字以内で、以下のいずれかの形式で記述します。

— yyyy-mm-dd

— yyyy.mm.dd

yyyyには年、mmには月、ddには日を数字で記述します。数字、「-」および「.」は、全角および半角のいずれでも記述できます。

記述例

TRANS, 取引高の累積, V12L10, 富士通, 2018-03-01

3.2.2 著作権表示行

著作権表示行は、モジュールの著作権を主張するために、著作権の所在を記述するところです。

記述形式

◆ 著作権表示

記述規則

著作権表示は、全角または半角の英字、数字、特殊文字および空白で記述します。

著作権表示行は継続することができません。

記述例

◆ COPYRIGHT 1990-2018 FUJITSU LIMITED

3.2.3 翻訳指示行

翻訳指示行は、YPS仕様書の解釈に関する指示を記述するところです。

記述形式

◇ 翻訳：指示の並び

記述規則

翻訳指示行の本文には、“表3.1 翻訳指示行に記述する指示一覧”で示した指示を、全角または半角の「,」で区切って記述します。区切り文字の前後には、全角または半角の空白を入れることもできます。また、指示中の「(,「)」、「1」および「2」は、全角でも半角でも記述できます。

翻訳指示行は継続することができません。したがって、指示の並びが一行に収まらない場合には、指示の並びを分割し、翻訳指示行を複数行記述します。翻訳指示行を複数行記述する場合には、各行の終わりには区切り文字は記述しません。

指示の一覧

翻訳指示行に記述する指示の一覧を“表3.1 翻訳指示行に記述する指示一覧”に示します。

表3.1 翻訳指示行に記述する指示一覧

指示	意味
言語仕様 ($\left(\begin{array}{c} 1 \\ 2 \end{array} \right)$)	どの言語仕様に基づいて記述した仕様書かを指示します。 <ul style="list-style-type: none">• 1 「FUJITSU YPS/COBOL言語文法書 1991年10月版」以前の言語仕様に基づいて記述されています。 「言語仕様(1)」を指示した場合には、旧言語仕様に関する他の指示も指定することができます。詳細は、“付録G 翻訳指示行に記述する旧仕様に関する指示”を参照してください。• 2 「FUJITSU YPS/COBOL言語文法書 1992年12月版」以降の言語仕様に基づいて記述されています。
引用符 ($\left(\begin{array}{c} \text{単} \\ \text{複} \end{array} \right)$)	仕様書内で使用している引用符の種類を指示します。 <ul style="list-style-type: none">• 単 アポストロフィ(')を使用しています。• 複 ダブルクォーテーション(")を使用しています。
全角文字記述 ($\left(\begin{array}{c} \text{あり} \\ \text{なし} \end{array} \right)$)	仕様書内で全角文字記述機能を使用しているかどうかを指示します。 <ul style="list-style-type: none">• あり 全角文字記述機能を使用しています。• なし 全角文字記述機能は使用していません。
行連結 ($\left(\begin{array}{c} \text{あり} \\ \text{なし} \end{array} \right)$)	継続行の連結方法を指示します。 <ul style="list-style-type: none">• あり 各行の末尾と継続行の先頭の空白を無視し、各行の間に半角の空白を1文字追加して連結します。• なし 記述した状態のまま先行する行と継続行を連結します。

記述例

◇翻訳：言語仕様（2），引用符（複），全角文字記述（あり）

一般規則

翻訳指示行に同じ指示が複数記述されていた場合には、後に記述した指示が有効になります。

翻訳指示行に記述する指示と等価な翻訳オプションがある場合には、翻訳指示行に記述した指示が優先されます。各指示と等価な翻訳オプションについては、各システムの使用手引書を参照してください。

3.2.4 モジュール概要記述

モジュール概要記述は、モジュールの機能やインターフェースなどの情報を記述するところです。記述した情報はプログラムの意味には何の影響も与えません。

記述規則

モジュールの機能やインターフェースなどの情報を自由に記述します。

YPS仕様書で使用できる文字で記述します。

記述例

```
【機能】
入力ファイルに入っているレコード（顧客名と取引高の組）を全件読み込み、
同じ顧客名の取引高を累積して一つのレコードにして、出力ファイルに書き出す。
ただし、当モジュールの仕様は、以下のとおりとする。
- 顧客数は、500社以下である。
- 総取引高は、8桁以下の値である。
```

3.3 モジュール本体を構成する表記

モジュール本体を構成する表記には、以下の3つの表記があります。

- ・ 構文定義部
- ・ モジュール共通データ部
- ・ プログラム単位

3.3.1 構文定義部

構文定義部は、抽象化処理構文、抽象化条件式構文およびコメントキーワードを定義するところです。

構文定義部の範囲は、構文定義部開始からデータ部開始の前までです。モジュール共通データ部を省略した場合には、構文定義部開始からプログラム開始の前までです。

抽象化処理構文および抽象化条件式構文については、“[第7章 構文定義機能](#)”を、コメントキーワードについては、“[第6章 コメントキーワード定義機能](#)”を参照してください。

記述形式



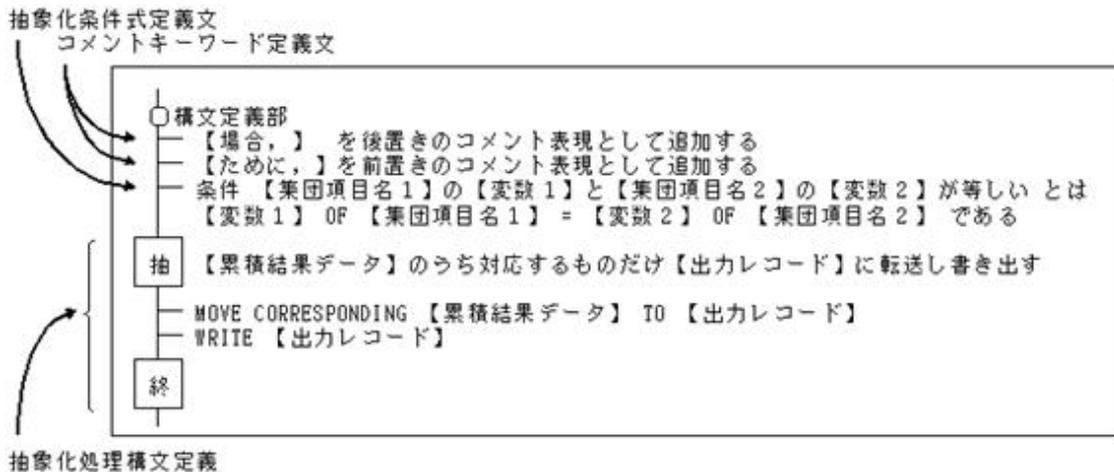
記述規則

構文定義部は、モジュール本体開始図記号の直後の主制御線上に記述します。

構文定義部には、抽象化処理構文定義、抽象化条件式定義文およびコメントキーワード定義文を任意の順序で記述します。

構文定義部は省略することができます。

記述例



3.3.2 モジュール共通データ部

モジュール共通データ部は、日本語名標宣言文を記述するところです。

モジュール共通データ部の範囲は、データ部開始からプログラム開始の前までです。

日本語名標宣言文については、“[第8章 日本語名標宣言機能](#)”を参照してください。

記述形式



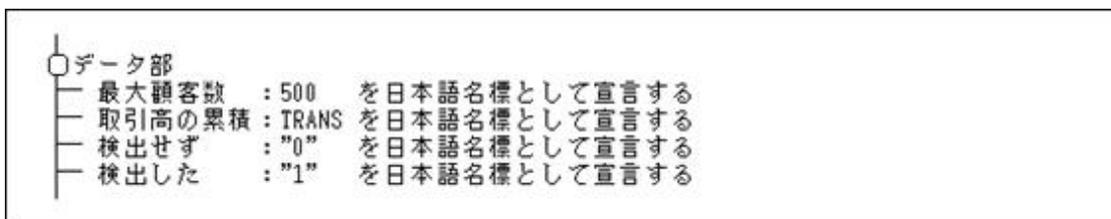
記述規則

モジュール共通データ部は、構文定義部の直後の主制御線上に記述します。構文定義部を省略した場合には、モジュール本体開始の直後の主制御線上に記述します。

モジュール共通データ部には、日本語名標宣言文を記述します。

モジュール共通データ部は省略することができます。

記述例

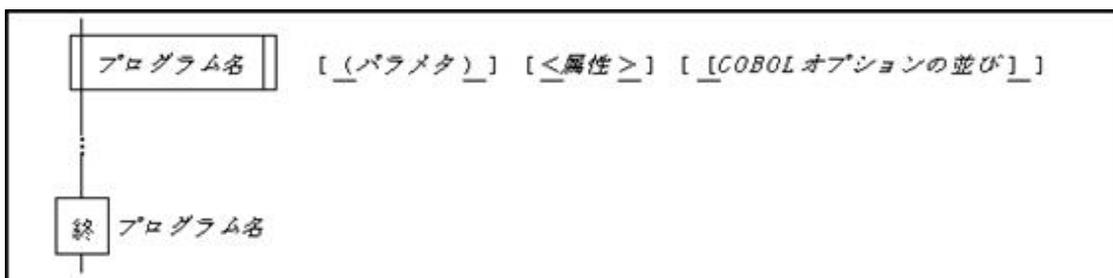


3.3.3 プログラム単位

プログラム単位は、COBOLの原始プログラムに対応する表記です。

プログラム単位の範囲は、プログラム開始からプログラム終了までです。

記述形式



記述形式に関する記述規則

プログラム開始図記号の中には、プログラム名を記述します。

プログラム開始図記号の本文には、パラメタ、属性およびCOBOLオプションの並びを任意の順序で記述します。また、パラメタ、属性およびCOBOLオプションの並びは、どれも省略することができます。

パラメタおよび属性を囲むシステム定義のキーワード「(、)」、「<」および「>」は、全角または半角で記述します。COBOLオプションの並びを囲むシステム定義のキーワード「[」および「]」は全角で記述します。

終了図記号の本文には、プログラム開始図記号の中に記述したプログラム名と同じプログラム名を記述します。

プログラム名

プログラム名には、COBOLのプログラム名を記述します。

パラメタ

パラメタは、変数の受け渡しを行う場合に指定します。

パラメタには、プログラム単位が受け取る変数を、COBOLのUSING指定の中の作用対象の記述規則に従って記述します。ただし、語USINGは指定する必要はありません。

プログラム開始の本文に記述したパラメタは、プログラム内データ部の仮引数宣言節で宣言します。

属性

属性は、プログラム単位が副プログラムの場合に指定します。

属性には、初期化プログラムであることを表すCOBOLの予約語「INITIAL」を指定することができます。

COBOLオプションの並び

COBOLオプションの並びは、COBOL原始プログラムの翻訳時にCOBOLコンパイラに指示を与える場合に記述します。

COBOLオプションの並びには、COBOL原始プログラム翻訳時の翻訳オプションを記述します。指定できる翻訳オプションの名前、オプションの意味およびオプションの指定方法は、各システムのCOBOLの使用手引書を参照してください。



参照

COBOLの用語

原始プログラム

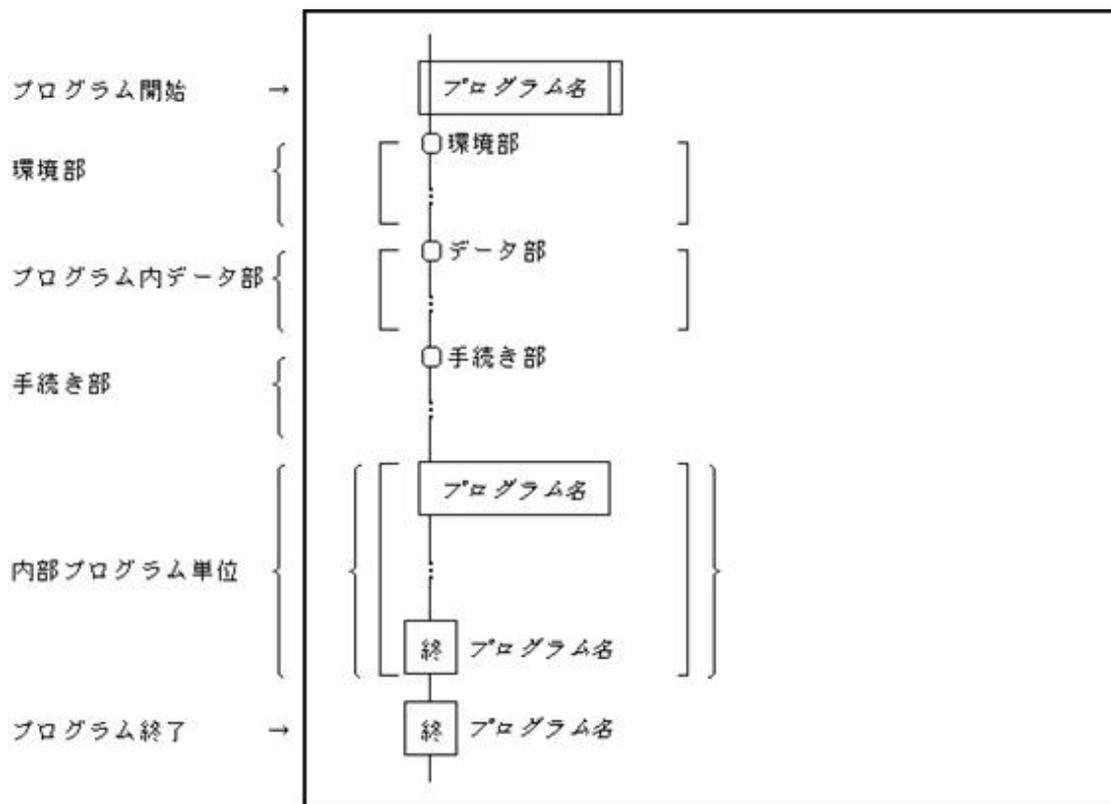
プログラム名

USING指定 ⇒ 手続き部の見出し

副プログラム

初期化プログラム

構成



構成に関する記述規則

プログラム単位内には、環境部、プログラム内データ部、手続き部および内部プログラム単位を構成で示した順に記述します。

環境部およびデータ部は省略することができますが、手続き部は省略することはできません。

プログラム単位には、1つ以上の内部プログラム単位が記述できます。内部プログラム単位は記述しなくてもかまいません。

記述例



3.4 プログラム単位を構成する表記

プログラム単位を構成する表記には、以下の4つの表記があります。

- ・ 環境部
- ・ プログラム内データ部

- ・ 手続き部
- ・ 内部プログラム単位

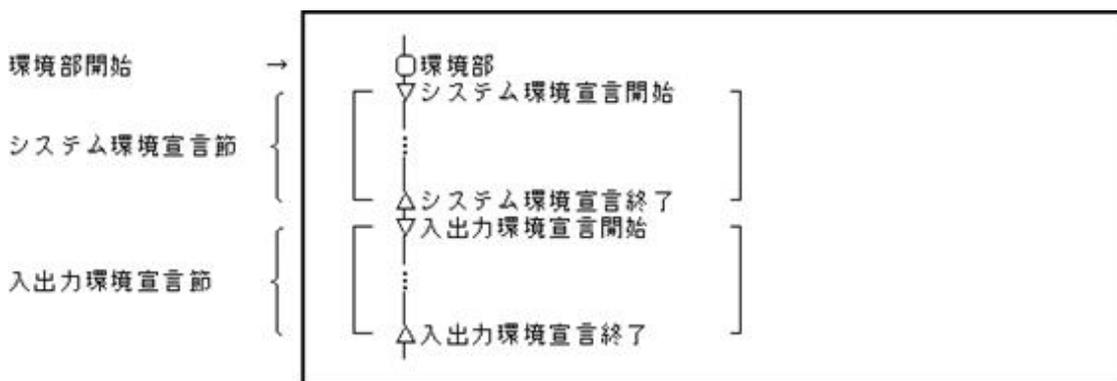
3.4.1 環境部

環境部は、翻訳時または実行時の計算機の環境、および実行時に使用するファイルの情報を記述するところです。

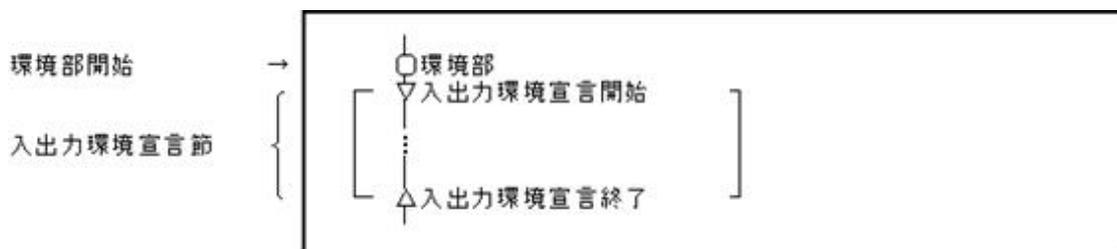
環境部の範囲は、環境部開始からデータ部開始の前までです。プログラム内データ部を省略した場合には、環境部開始から手続き部開始の前までです。

構成

[プログラム単位の場合]



[内部プログラム単位の場合]



記述規則

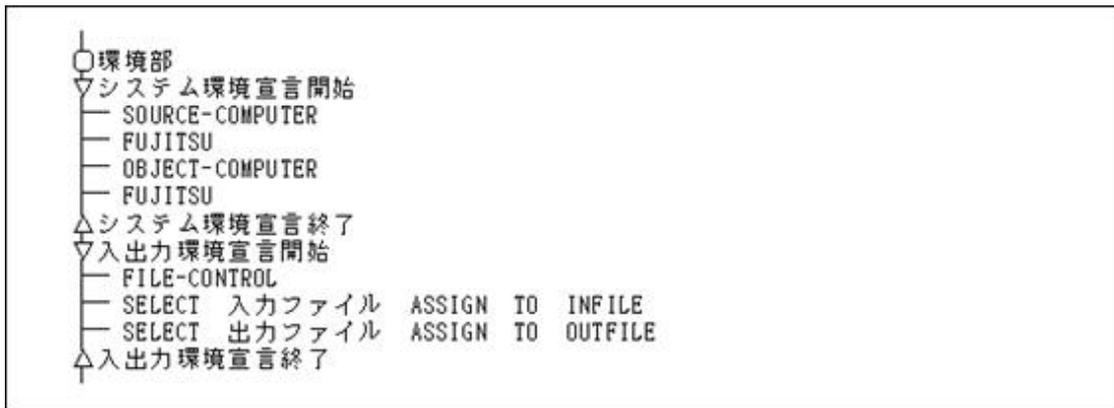
環境部は、プログラム開始または内部プログラム開始の直後の主制御線上に記述します。

プログラム単位内の環境部には、システム環境宣言節および入出力環境宣言節を構成で示した順序で記述します。

内部プログラム単位の環境部には、入出力環境宣言節を記述します。

システム環境宣言節および入出力環境宣言節は省略することができます。

記述例

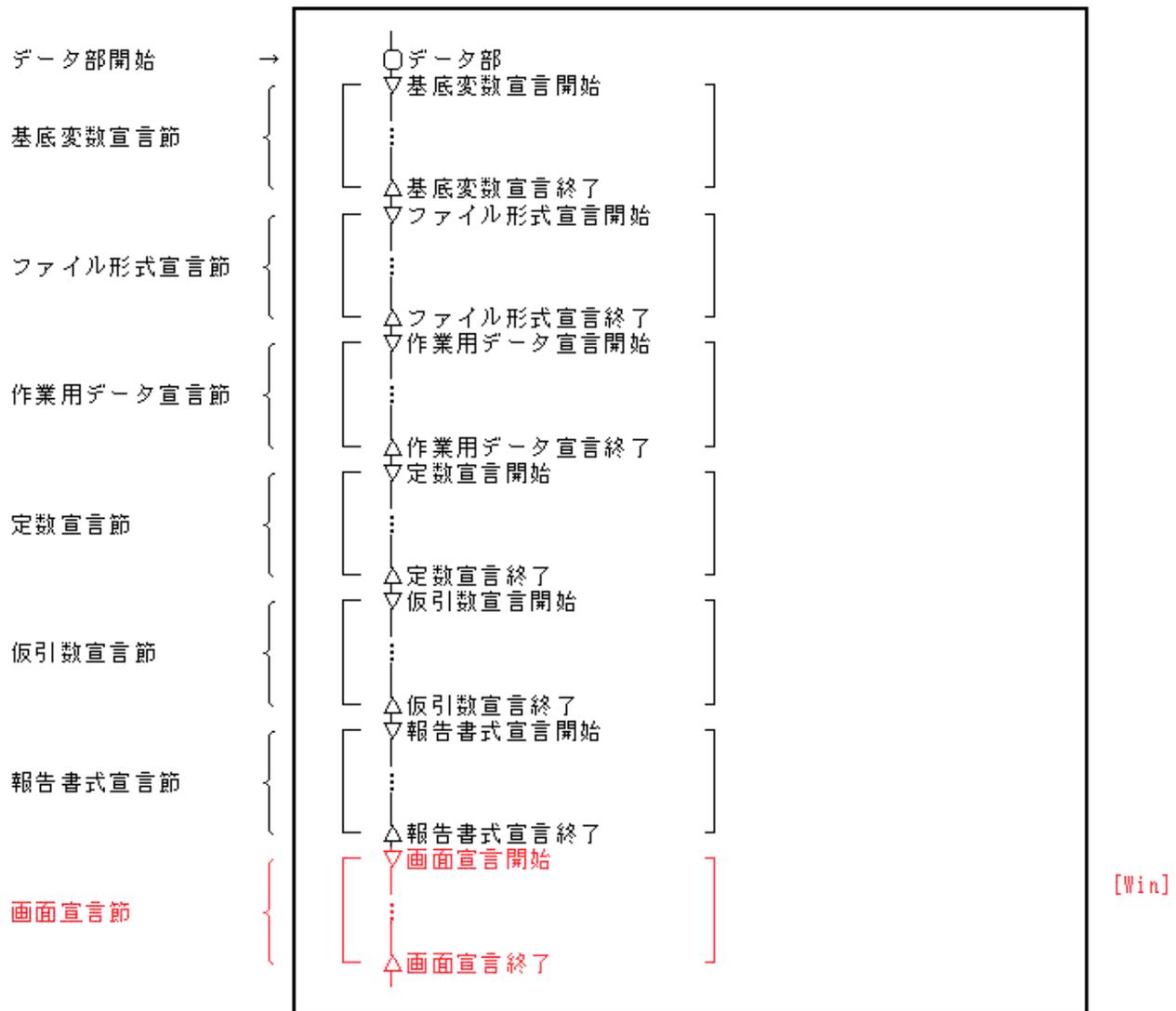


3.4.2 プログラム内データ部

プログラム内データ部は、プログラム単位および内部プログラム単位で使用するデータを宣言するところです。日本語名標を宣言することもできます。

プログラム内データ部の範囲は、データ部開始から手続き部開始の前までです。

構成

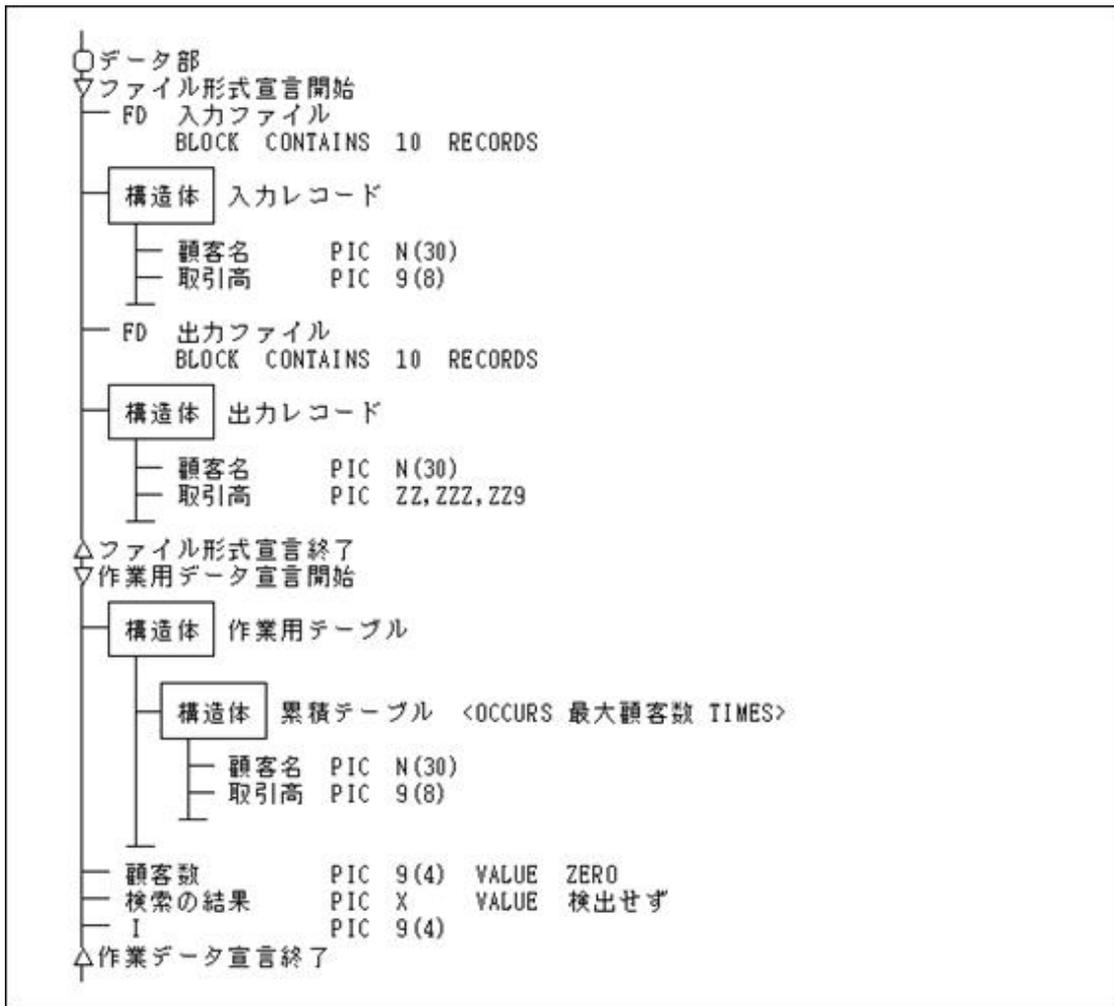


プログラム内データ部は、環境部の直後の主制御線上に記述します。環境部を省略した場合には、プログラム単位または内部プログラム単位の直後の主制御線上に記述します。

プログラム内データ部には、基底変数宣言節、ファイル形式宣言節、作業用データ宣言節、定数宣言節、仮引数宣言節、報告書式宣言節 [Win]および画面宣言節を構成で示した順序で記述します。

プログラム内データ部に記述する各節は省略することができます。

記述例



3.4.3 手続き部

手続き部は、プログラムの処理手続きを記述するところです。

プログラム単位における手続き部の範囲は、手続き部開始からプログラム終了の前または内部プログラム開始の前までです。内部プログラム単位における手続き部の範囲は、手続き部開始から内部プログラム終了の前または内部プログラム開始の前までです。

構成



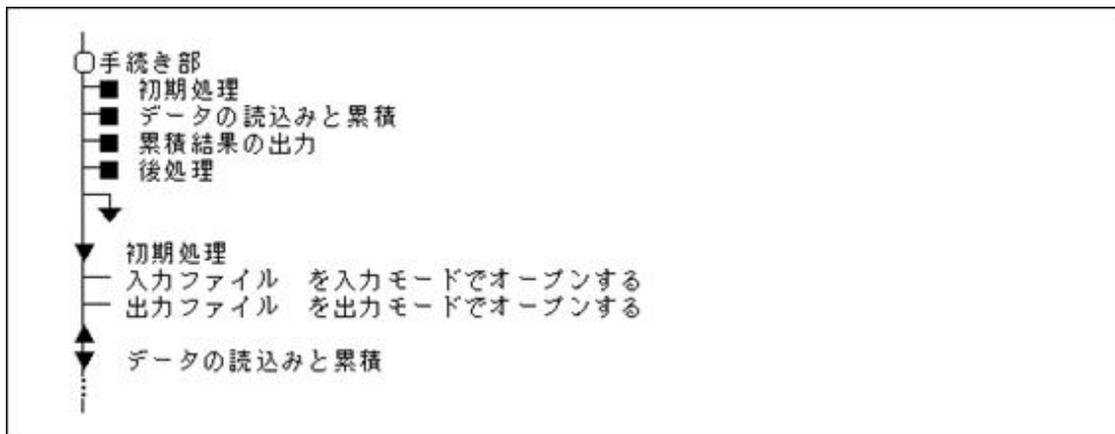
記述規則

手続き部は、プログラム単位または内部プログラム単位内の主制御線上に記述します。

手続き部内には、宣言部分と手続き部分を構成に示した順序で記述します。宣言部分は省略することができます。

手続き部分には、1つ以上の節を記述します。最初の節は、節開始と節終了を省略することができます。宣言部分の記述規則については、“3.9.1 宣言部分”を参照してください。

記述例

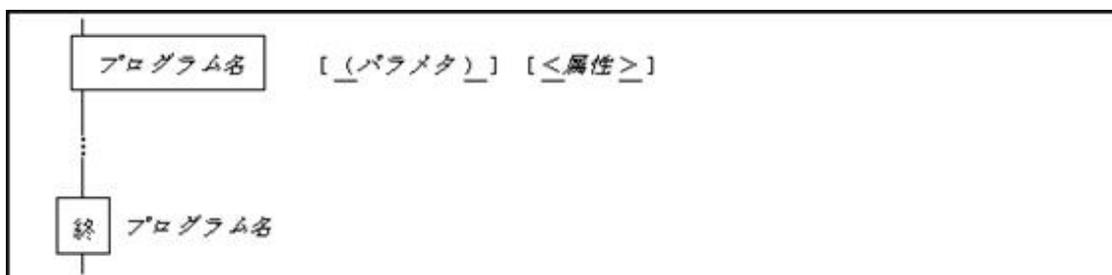


3.4.4 内部プログラム単位

内部プログラム単位は、COBOLの含まれるプログラムに対応する表記です。

内部プログラム単位は、内部プログラム開始から内部プログラム終了までの範囲をいいます。

記述形式



記述形式に関する記述規則

内部プログラム開始図記号の中には、プログラム名を記述します。

内部プログラム開始図記号の本文には、パラメタおよび属性を任意の順序で記述します。また、パラメタおよび属性は省略することができます。

パラメタおよび属性を囲むYPSのキーワード「(」、「)」、「<」および「>」は、全角または半角で記述します。

終了図記号の本文には、内部プログラム開始図記号の中に記述したプログラム名と同じプログラム名を記述します。

プログラム名

プログラム名には、COBOLのプログラム名を記述します。

パラメタ

パラメタは、変数の受け渡しを行う場合に指定します。

パラメタには、内部プログラム単位が受け取る変数を、COBOLのUSING指定の中の作用対象の記述規則に従って記述します。ただし、語USINGは指定する必要はありません。

内部プログラム開始の本文に記述したパラメタは、プログラム内データ部の仮引数宣言節で宣言します。

属性

属性には、共通プログラムであることを表すCOBOLの予約語「COMMON」、または初期化プログラムであることを表すCOBOLの予約語「INITIAL」を指定することができます。

参照

COBOLの用語

含まれるプログラム ⇒ 原始プログラムの入れ子

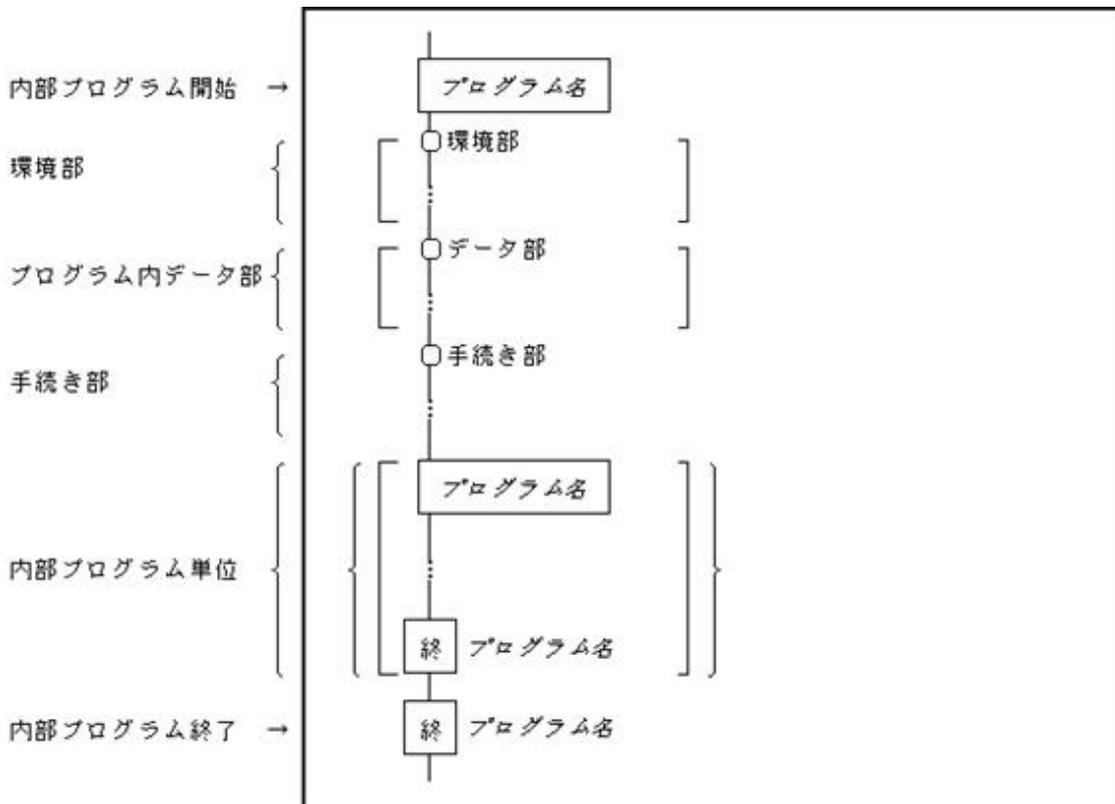
プログラム名

USING指定 ⇒ 手続き部の見出し

共通プログラム

初期化プログラム

構成



構成に関する記述規則

内部プログラム単位内には、環境部、プログラム内データ部、手続き部および内部プログラム単位を構成で示した順に記述します。

環境部およびデータ部は省略することができますが、手続き部は省略することはできません。

内部プログラム単位には1つ以上の内部プログラム単位を記述できます。内部プログラム単位は記述しなくてもかまいません。

記述例



3.5 環境部を構成する表記

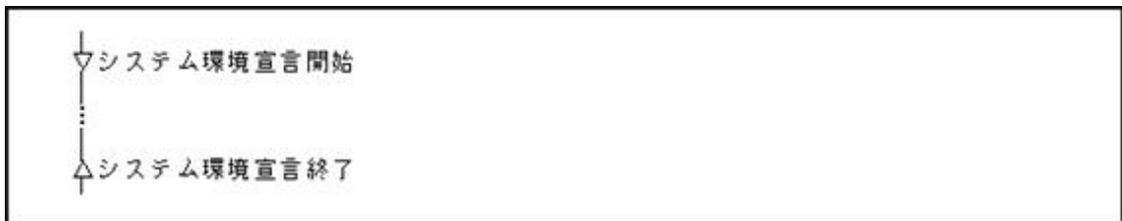
環境部を構成する表記には、以下の2つの表記があります。

- ・ システム環境宣言節
- ・ 入出力環境宣言節

3.5.1 システム環境宣言節

システム環境宣言節は、翻訳時および実行時の計算機の環境を宣言するところです。COBOLの構成節に対応する表記です。

記述形式



記述規則

システム環境宣言節は、プログラム単位の環境部の主制御線上に記述します。

システム環境宣言節は省略することができます。

システム環境宣言節には、以下に示すCOBOLの段落を宣言文で記述します。記述方法については“3.8.1 宣言文”を参照してください。

- ・ 翻訳用計算機段落
- ・ 実行用計算機段落
- ・ 特殊名段落



参照

COBOLの用語

構成節

翻訳用計算機段落

実行用計算機段落

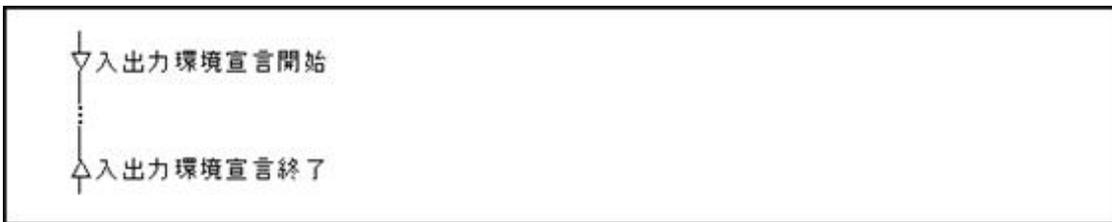
記述例



3.5.2 入出力環境宣言節

入出力環境宣言節は、実行時に使用するファイルの情報を記述するところです。COBOLの入出力節に対応する表記です。

記述形式



記述規則

入出力環境宣言節は、環境部の主制御線上に記述します。

入出力環境宣言節は省略することができます。

入出力環境宣言節には、以下に示すCOBOLの段落を宣言文で記述します。記述方法については“[3.8.1 宣言文](#)”を参照してください。

- ・ ファイル管理段落
- ・ 入出力管理段落



参照

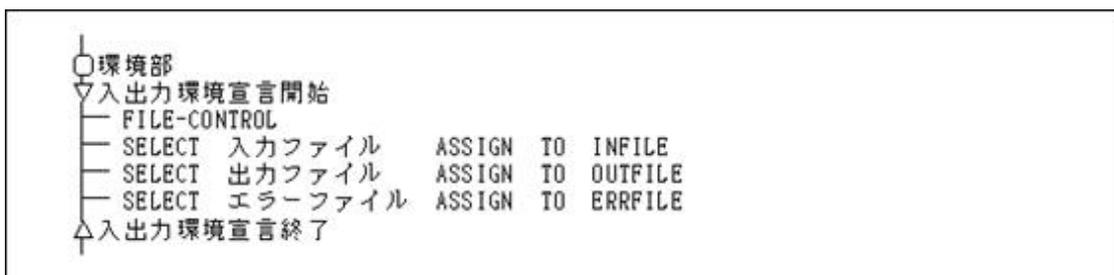
COBOLの用語

入出力節

ファイル管理段落

入出力管理段落

記述例



3.6 環境部の節を構成する表記

環境部の節を構成する表記は宣言文だけです。宣言文については“[3.8.1 宣言文](#)”を参照してください。

3.7 プログラム内データ部を構成する表記

プログラム内データ部を構成する表記には、以下の7つの表記があります。

- 基底変数宣言節
- ファイル形式宣言節
- 作業用データ宣言節
- 定数宣言節
- 仮引数宣言節
- 報告書式宣言節
- **[Win]** 画面宣言節

3.7.1 基底変数宣言節

基底変数宣言節は、ポインタデータ項目により、領域のアドレスが決定される変数を宣言するところです。COBOLの基底場所節に対応する表記です。

記述形式



記述規則

基底変数宣言節は、プログラム内データ部の主制御線上に記述します。

基底変数宣言節は省略することができます。

基底変数宣言節には、以下に示すCOBOLの記述項を宣言文または構造体で記述します。記述方法については“[3.8.1 宣言文](#)”および“[3.8.2 構造体](#)”を参照してください。

- 独立データ記述項
- レコード記述項

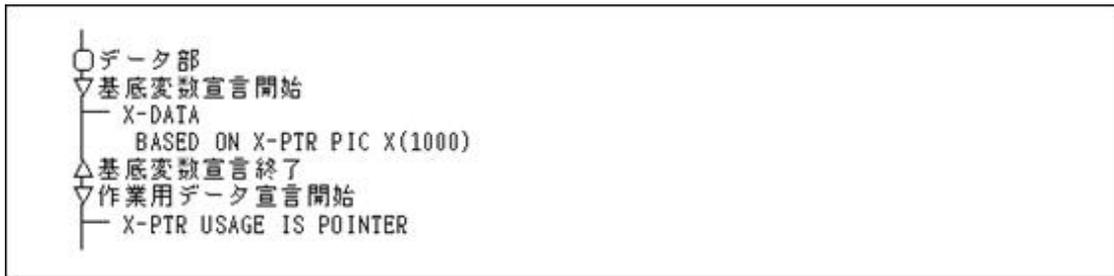


参照

COBOLの用語

- ポインタデータ項目
- 基底場所節
- 独立データ記述項
- レコード記述項

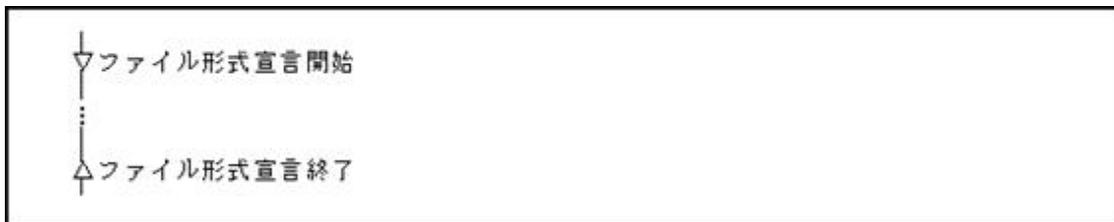
記述例



3.7.2 ファイル形式宣言節

ファイル形式宣言節は、実行時に画面宣言節使用するファイルの構造を宣言するところです。COBOLのファイル節に対応する表記です。

記述形式



記述規則

ファイル形式宣言節は、プログラム内データ部の主制御線上に記述します。

ファイル形式宣言節は省略することができます。

ファイル形式宣言節には、以下に示すCOBOLの記述項を宣言文または構造体で記述します。記述方法については“3.8.1 宣言文”および“3.8.2 構造体”を参照してください。

- ・ ファイル記述項
- ・ 整列併合用ファイル記述項
- ・ レコード記述項

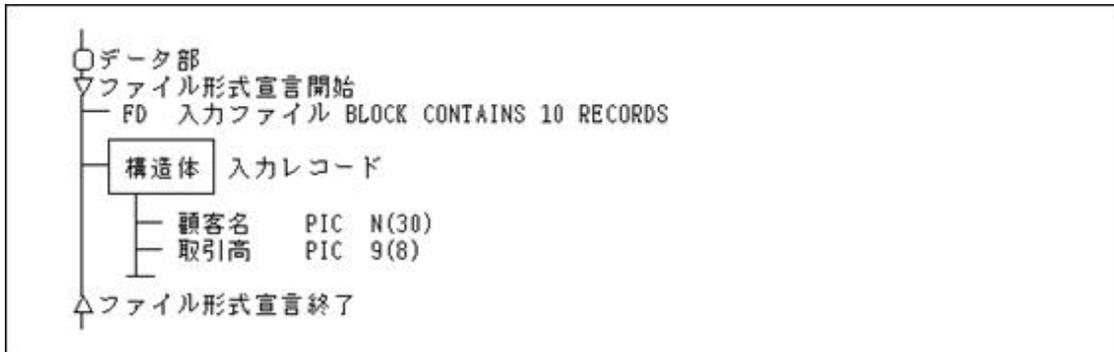


参照

COBOLの用語

- ファイル節
- ファイル記述項
- 整列併合用ファイル記述項

記述例



3.7.3 作業用データ宣言節

作業用データ宣言節は、プログラム単位内または内部プログラム単位内で使用する変数を宣言するところです。COBOLの作業場所節に対応する表記です。

記述形式



記述規則

作業用データ宣言節は、プログラム内データ部の主制御線上に記述します。

作業用データ宣言節は省略することができます。

作業用データ宣言節には、以下に示すCOBOLの記述項を宣言文または構造体で記述します。記述方法については“3.8.1 宣言文”および“3.8.2 構造体”を参照してください。

- ・ 独立データ記述項
- ・ レコード記述項

参照

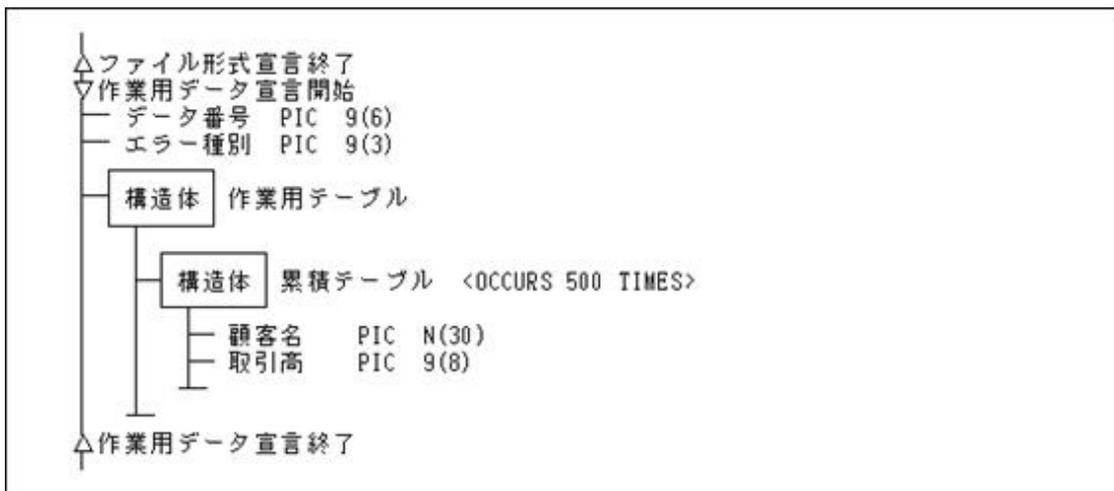
COBOLの用語

作業場所節

独立データ記述項

レコード記述項

記述例



3.7.4 定数宣言節

定数宣言節は、プログラムの実行中に値が変わらない変数を宣言するところです。COBOLの定数節に対応する表記です。

記述形式



記述規則

定数宣言節は、プログラム内データ部の主制御線上に記述します。

定数宣言節は省略することができます。

定数宣言節には、以下に示すCOBOLの記述項を宣言文または構造体で記述します。記述方法については“3.8.1 宣言文”および“3.8.2 構造体”を参照してください。

- ・ 独立データ記述項
- ・ レコード記述項



参照

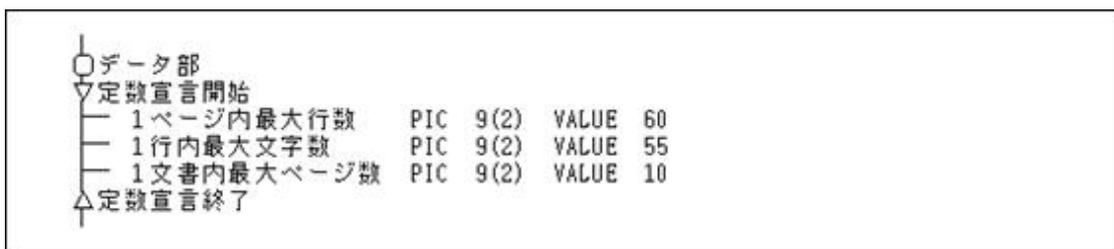
COBOLの用語

定数節

独立データ記述項

レコード記述項

記述例



3.7.5 仮引数宣言節

仮引数宣言節は、呼ぶプログラムから渡される変数を宣言するところです。COBOLの連絡節に対応する表記です。

記述形式



記述規則

仮引数宣言節は、プログラム内データ部の主制御線上に記述します。

仮引数宣言節は省略することができます。

仮引数宣言節には、以下に示すCOBOLの記述項を宣言文または構造体で記述します。記述方法については、“[3.8.1 宣言文](#)”および“[3.8.2 構造体](#)”を参照してください。

- ・ 独立データ記述項
- ・ レコード記述項

一般規則

プログラム開始および内部プログラム開始の本文にパラメタとして宣言した変数は、必ず仮引数宣言節で宣言してください。また、二次入口の本文にパラメタとして宣言した変数は、その二次入口が含まれるプログラム単位の仮引数宣言節で宣言してください。

参照

COBOLの用語

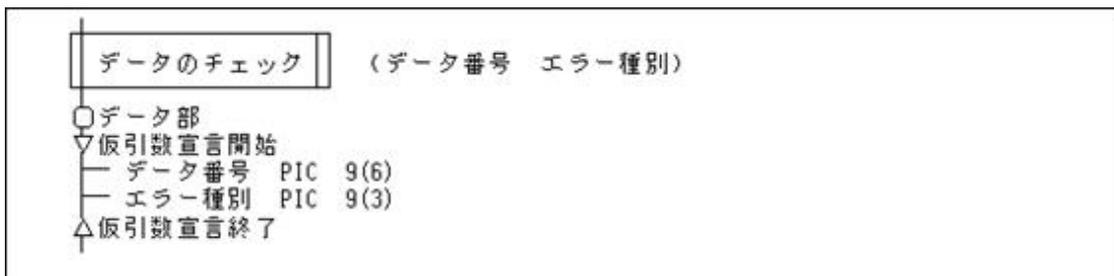
呼ぶプログラム

連絡節

独立データ記述項

レコード記述項

記述例

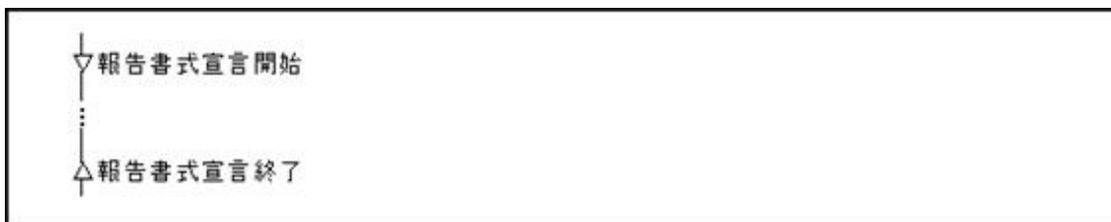


備考: プログラム名を日本語で記述する場合、日本語名標宣言文であらかじめ宣言しておく必要があります。

3.7.6 報告書式宣言節

報告書式宣言節は、報告書ファイルに書き出される報告書の書式を宣言するところです。COBOLの報告書節に対応する表記です。

記述形式



記述規則

報告書式宣言節は、プログラム内データ部の主制御線の上に記述します。

報告書式宣言節は省略することができます。

報告書式宣言節には、以下に示すCOBOLの記述項を宣言文または構造体で記述します。記述方法については“3.8.1 宣言文”および“3.8.2 構造体”を参照してください。

- ・ 報告書記述項
- ・ 報告集団記述項



参照

COBOLの用語

報告書ファイル

報告書節

報告書記述項

報告集団記述項

記述例



3.7.7 画面宣言節 [Win]

画面宣言節は、ディスプレイに表示する画面のフォーマットを定義するところです。COBOLの画面節に対応する表記です。

記述形式



記述規則

画面宣言節は、プログラム内データ部の主制御線上に記述します。

画面宣言節は省略することができます。

画面宣言節には、以下に示すCOBOLの記述項を宣言文または構造体で記述します。記述方法については“3.8.1 宣言文”および“3.8.2 構造体”を参照してください。

- 画面データ記述項



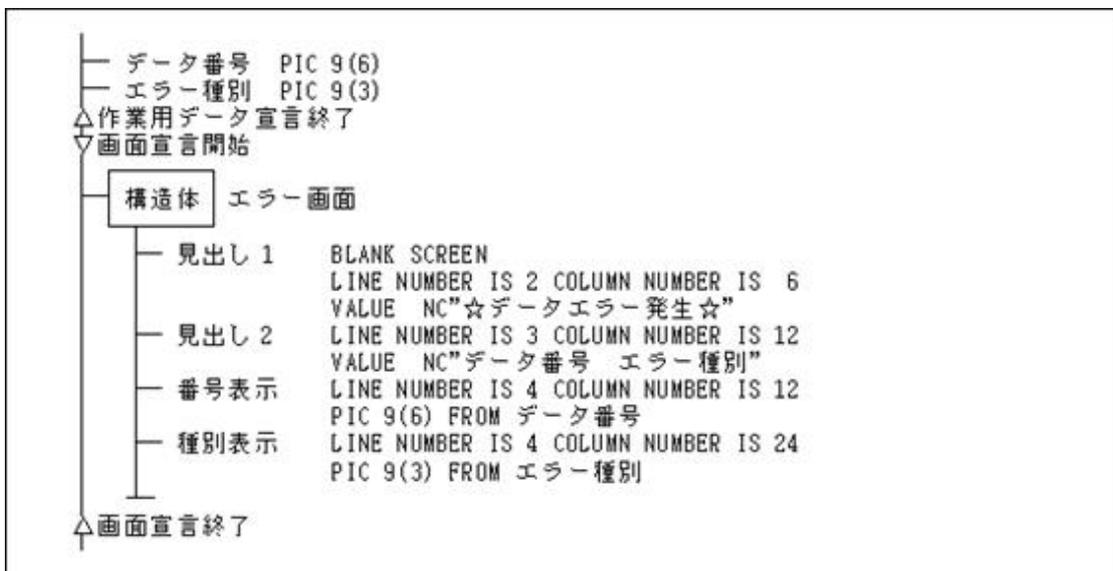
参照

COBOLの用語

画面節

画面データ記述項

記述例



3.8 データ部の節を構成する表記

プログラム内データ部の各節を構成する表記には、以下の2つの表記があります。

- 宣言文
- 構造体

3.8.1 宣言文

宣言文は、環境部の各節において環境を、プログラム内データ部の各節においてデータを宣言します。

記述形式



記述規則

宣言文は、環境部の節内およびデータ部の節内に記述します。

順次図記号の本文には、COBOLの文を記述します。

COBOLの文

COBOLの文には、以下に示すCOBOLの段落および記述項を記述します。

- ー 環境部の節内
 - 翻訳用計算機段落
 - 実行用計算機段落
 - 特殊名段落
 - ファイル管理段落
 - 入出力管理段落
- ー プログラム内データ部の節内
 - 下記を「データ記述項」と呼びます。
 - ファイル記述項
 - レコード記述項
 - 独立データ記述項
 - 報告書記述項
 - 報告集団記述項
 - [Win]画面データ記述項



注意

段落の見出しと記述項を一文で記述することはできません。

段落の見出しの終わりを表す終止符「.」、段落の終わりを表す終止符「.」および記述項の終わりを表す終止符「.」は、記述する必要はありません。

データ記述項には、以下の場合を除いてレベル番号を指定する必要はありません。

- ・ レベル番号66の記述項
- ・ レベル番号77の記述項
- ・ レベル番号88の記述項

COBOLの用語

- 段落
- 記述項
- データ記述項
- 段落見出し
- レベル番号

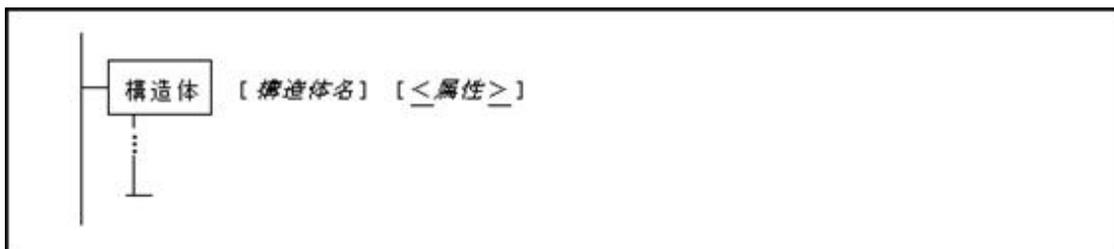
記述例



3.8.2 構造体

構造体は、階層構造を持つデータを宣言します。COBOLの集団項目に対応する表記です。

記述形式



記述規則

構造体は、データ部の節内に記述します。

構造体図記号の本文には、構造体名および属性を記述形式で示した順序で記述します。構造体名および属性は省略することができます。

属性を囲むシステム定義のキーワード「<」および「>」は、全角でも半角でも記述できます。

構造体内には、宣言文または構造体を1つ以上記述します。

構造体名

構造体名には、COBOLの集団項目名を記述します。

属性

属性には、COBOLの集団項目のデータ記述項に指定できるデータ句を記述します。

COBOLの用語

- 集団項目
- 集団項目名
- データ記述項
- データ句

記述例



3.9 手続き部を構成する表記

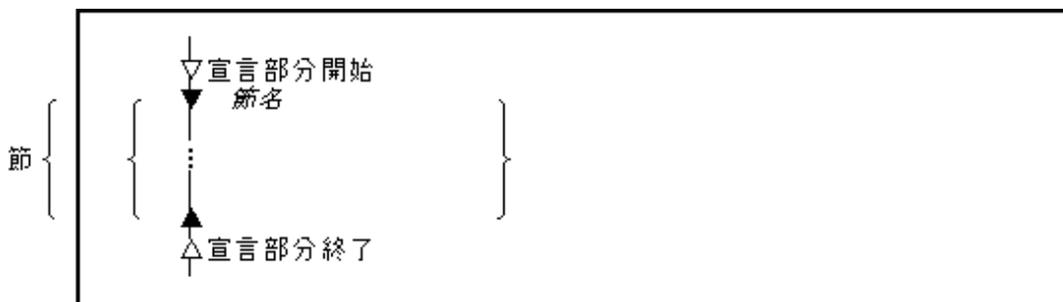
手続き部を構成する表記には、以下の2つの表記があります。

- ・ 宣言部分
- ・ 節

3.9.1 宣言部分

宣言部分は、例外事象が発生した場合の処理を記述するところです。COBOLの宣言部分(DECLARATIVES)に対応する表記です。

記述形式



記述規則

宣言部分は、手続き部開始の直後の主制御線上に記述します。

宣言部分は省略することができます。

宣言部分には、1つ以上の節を記述します。

宣言部分に記述する節には、COBOLのUSE文で始まる手続きを記述します。

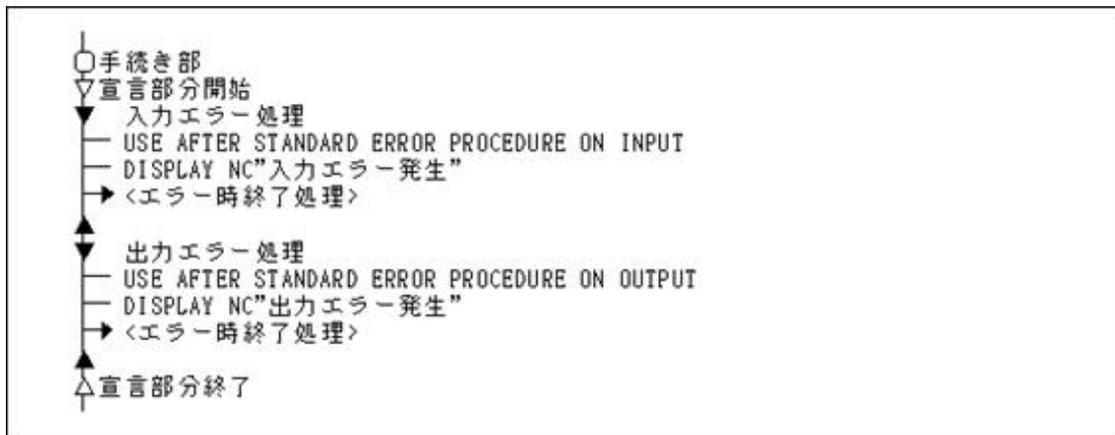
参照

COBOLの用語

宣言部分(DECLARATIVES)

USE文

記述例



3.9.2 節

節は、ひとまとまりの処理を記述するところです。COBOLの節に対応する表記です。

記述形式



記述規則

節は、宣言部分および手続き部分の主制御線上に記述します。手続き部分に記述する場合、最初の節は節開始および節終了を省略することができます。

節開始図記号の本文には、節名を記述します。

節には、“[3.10 手続き部の節を構成する表記](#)”で示す表記を記述します。

節名

節名には、COBOLの節名を記述します。

一般規則

節には直前の節から制御が渡されます。また、節呼出し表記で制御を渡すこともできます。節呼出しについては“3.10.6.7 節呼出し”を参照してください。

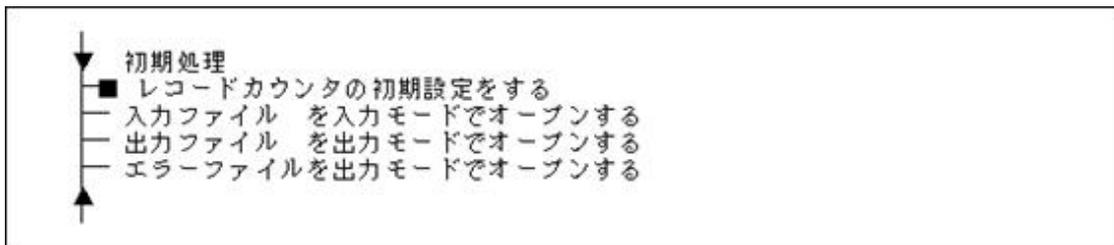


COBOLの用語

節

節名

記述例



3.10 手続き部の節を構成する表記

手続き部の節を構成する表記には、以下の6つの表記があります。

- ・ 順次系の表記
- ・ 選択系の表記
- ・ ループ系の表記
- ・ 例外処理
- ・ 検索処理
- ・ 制御変更系の表記

3.10.1 順次系の表記

順次系の表記には、以下の2種類があります。

- ・ 順次文
- ・ 空文

3.10.1.1 順次文

順次文は、記述された順に処理を実行します。

記述形式



記述規則

順次図記号の本文には、COBOLの文を記述します。完結文の終わりを表す終止符「.」は記述しません。

COBOLの文

COBOLの文とは、COBOLの言語仕様で規定される文のことをいいます。



COBOLの用語

文

完結文

記述例

```
├ DISPLAY NC”入力データに誤りがあります。”
```

3.10.1.2 空文

空文は、何の処理も行いません。

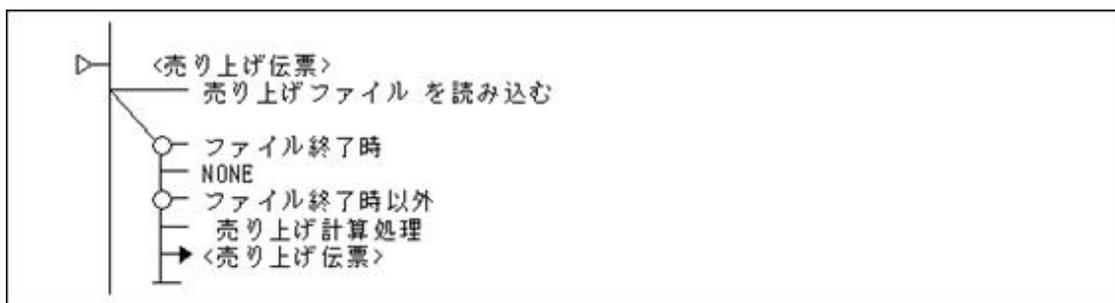
記述形式

```
├ NONE
```

記述規則

順次図記号の本文に、システム定義のキーワード「NONE」を全角または半角で記述します。

記述例



3.10.2 選択系の表記

選択系の表記には、以下の6種類があります。

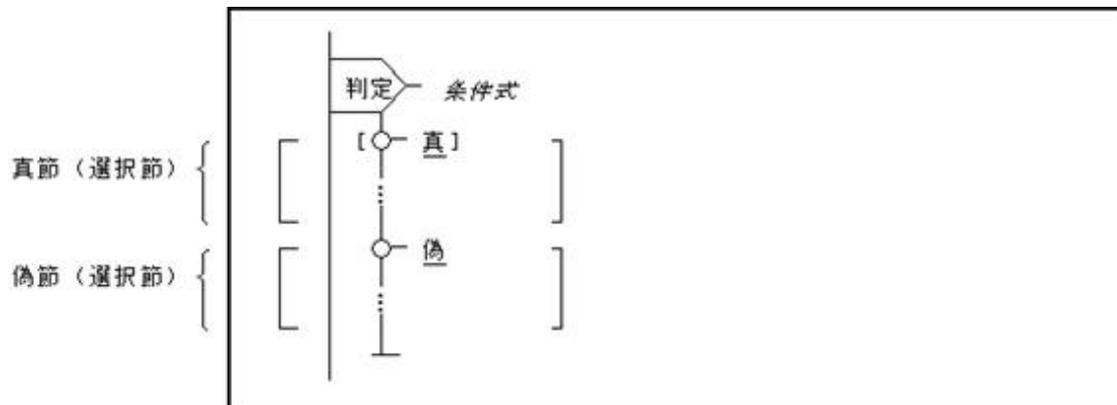
- ・ 判定
- ・ 単一選択(定数指定)
- ・ 単一選択(条件式指定)
- ・ 複合選択(定数指定)
- ・ 複合選択(条件式指定)
- ・ 評価

3.10.2.1 判定

判定は、条件式を評価し、その評価結果により実行する処理を選択します。

条件式が真の場合には、真節があれば真節の処理を実行し、なければ何も実行しません。条件式が偽の場合には、偽節があれば偽節の処理を実行し、なければ何も実行しません。

記述形式



記述規則

判定図記号の本文には、条件式を記述します。記述規則については“7.3.3 条件式”を参照してください。

判定には、真節と偽節を記述形式に示した順序で記述します。真節または偽節のどちらか一方を省略することができます。

真節

真節は、条件式の評価結果が真の場合に行う処理を記述するところです。

真節の開始を表す丸図記号の本文には、システム定義のキーワード「真」、「YES」または「yes」を記述します。「YES」および「yes」は全角でも半角でも記述できます。

偽節を省略する場合には、真節の開始を表す「○ 真」を省略することができます。

真節には、手続き部の節を構成する表記を1つ以上記述します。

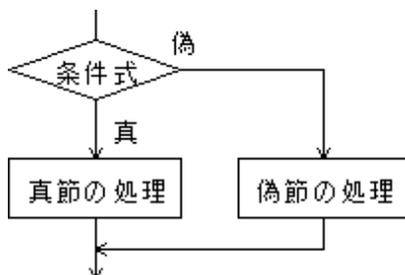
偽節

偽節は、条件式の評価結果が偽の場合に行う処理を記述するところです。

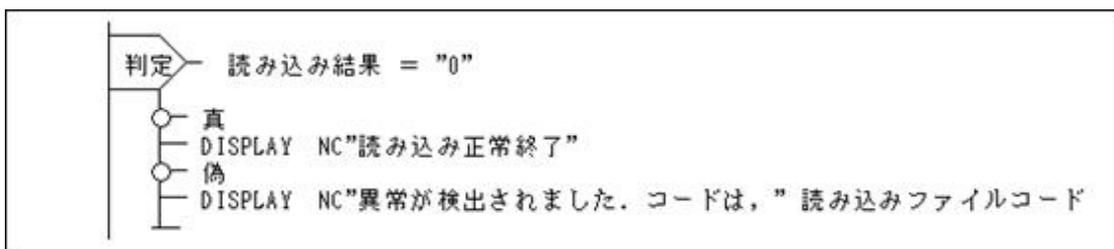
偽節の開始を表す丸図記号の本文には、システム定義のキーワード「偽」、「NO」または「no」を記述します。「NO」および「no」は全角でも半角でも記述できます。

偽節には、手続き部の節を構成する表記を一つ以上記述します。

処理の流れ



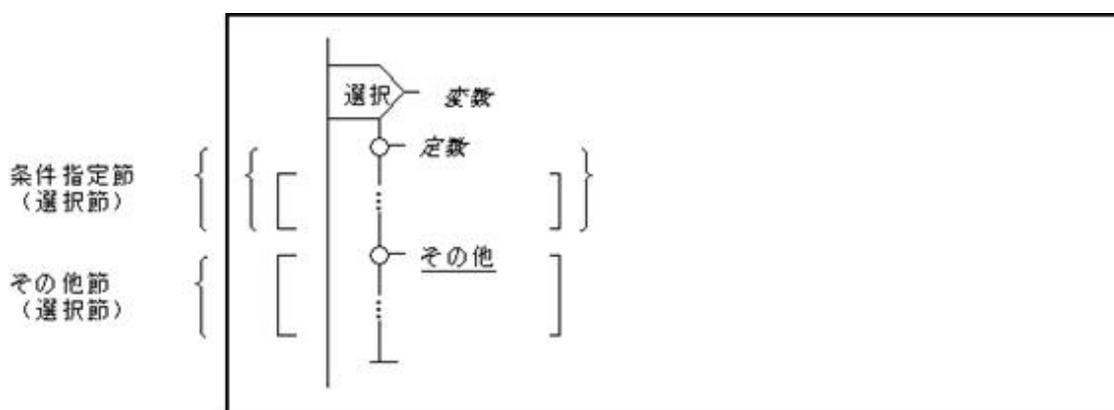
記述例



3.10.2.2 単一選択(定数指定)

単一選択(定数指定)は、変数と定数を上から比較して、最初に値が一致した条件指定節の処理を実行します。値が一致する条件指定節がなかった場合には、その他節があればその他節の処理を実行し、なければ何も実行しません。

記述形式



記述規則

選択図記号の本文には、変数を記述します。

単一選択(定数指定)には、1つ以上の条件指定節とその他節を記述形式で示した順序で記述します。その他節は省略することができます。

変数

変数には、COBOLの一意名を記述します。

条件指定節

条件指定節は、変数と定数の値が一致したときの処理を記述するところです。

条件指定節の開始を表す丸図記号の本文には、定数を記述します。

条件指定節には、手続き部の節を構成する表記を記述します。

定数

定数には、COBOLの定数を記述します。一意名と定数の項類は比較可能な組合せでなければなりません。

その他節

その他節は、値が一致する条件指定節がなかったときの処理を記述するところです。

その他節の開始を表す丸図記号の本文には、システム定義のキーワード「その他」、「OTHER」または「other」を記述します。「OTHER」および「other」は全角でも半角でも記述できます。

その他節には、手続き部の節を構成する表記を1つ以上記述します。

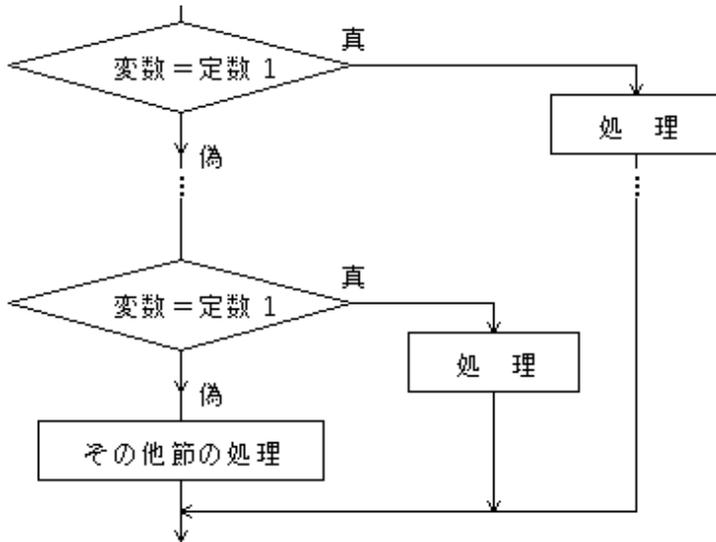
一般規則

条件指定節に、手続き部の節を構成する表記を1つも記述しなかった場合には、空文が省略されているものとみなされます。

COBOLの用語

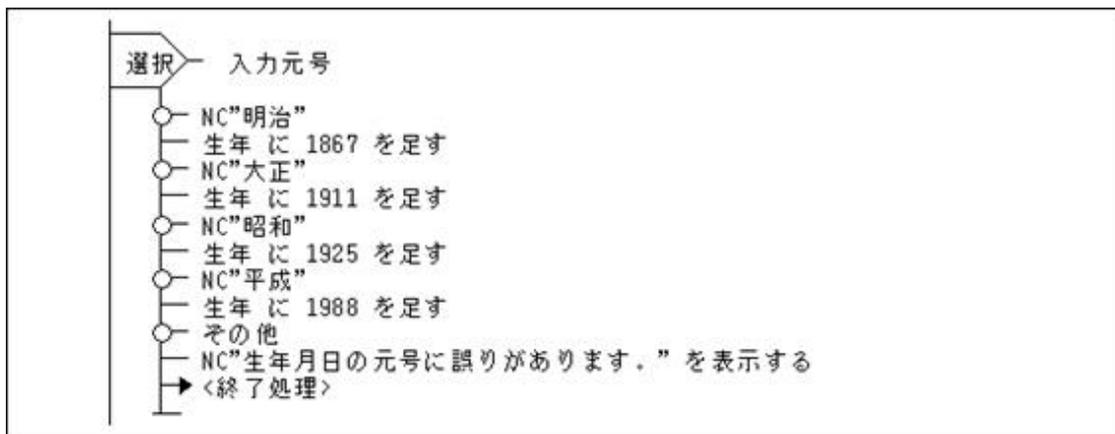
- 一意名
- 定数
- 項類

処理の流れ



備考:「処理」は、条件指定節に記述された処理を表しています。

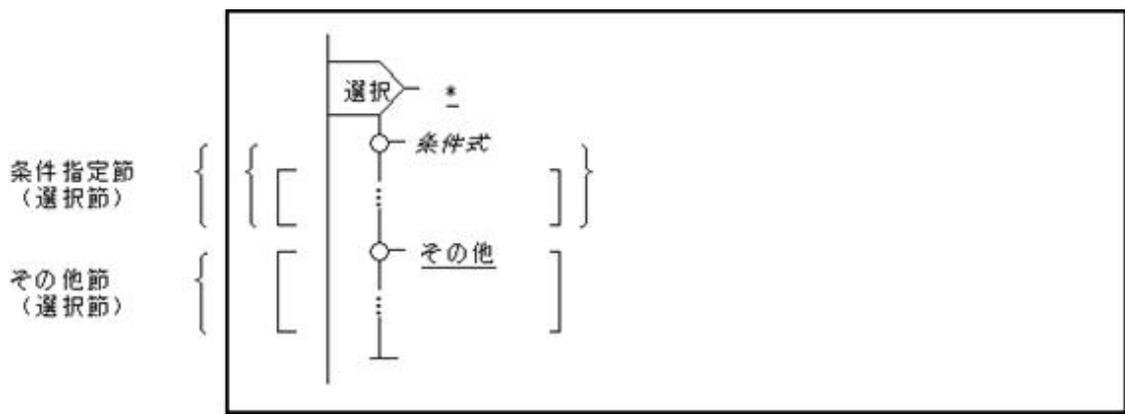
記述例



3.10.2.3 単一選択(条件式指定)

単一選択(条件式指定)は、条件式を上から評価し、最初に条件式が真になった条件指定節の処理を実行します。条件式が真である条件指定節がなかった場合には、その他節があればその他節の処理を実行し、なければ何も実行しません。

記述形式



記述規則

選択図記号の本文には、システム定義のキーワード「*」を記述します。「*」は全角でも半角でも記述できます。

単一選択(条件式指定)には、1つ以上の条件指定節とその他節を記述形式で示した順序で記述します。その他節は省略することができます。

条件指定節

条件指定節は、条件式が真になったときの処理を記述するところです。

条件指定節の開始を表す丸図記号の本文には、条件式を記述します。記述規則については“7.3.3 条件式”を参照してください。

条件指定節には、手続き部の節を構成する表記を記述します。

その他節

その他節は、条件式が真になる条件指定節がなかったときの処理を記述するところです。

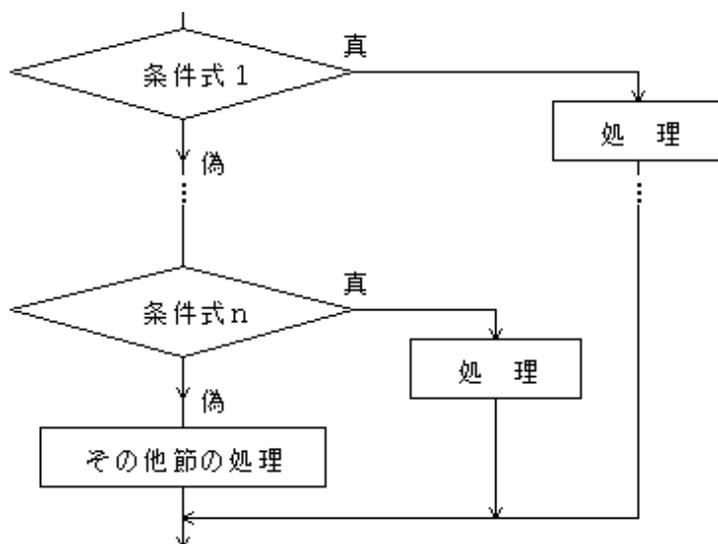
その他節の開始を表す丸図記号の本文には、システム定義のキーワード「その他」、「OTHER」または「other」を記述します。「OTHER」および「other」は全角でも半角でも記述できます。

その他節には、手続き部の節を構成する表記を1つ以上記述します。

一般規則

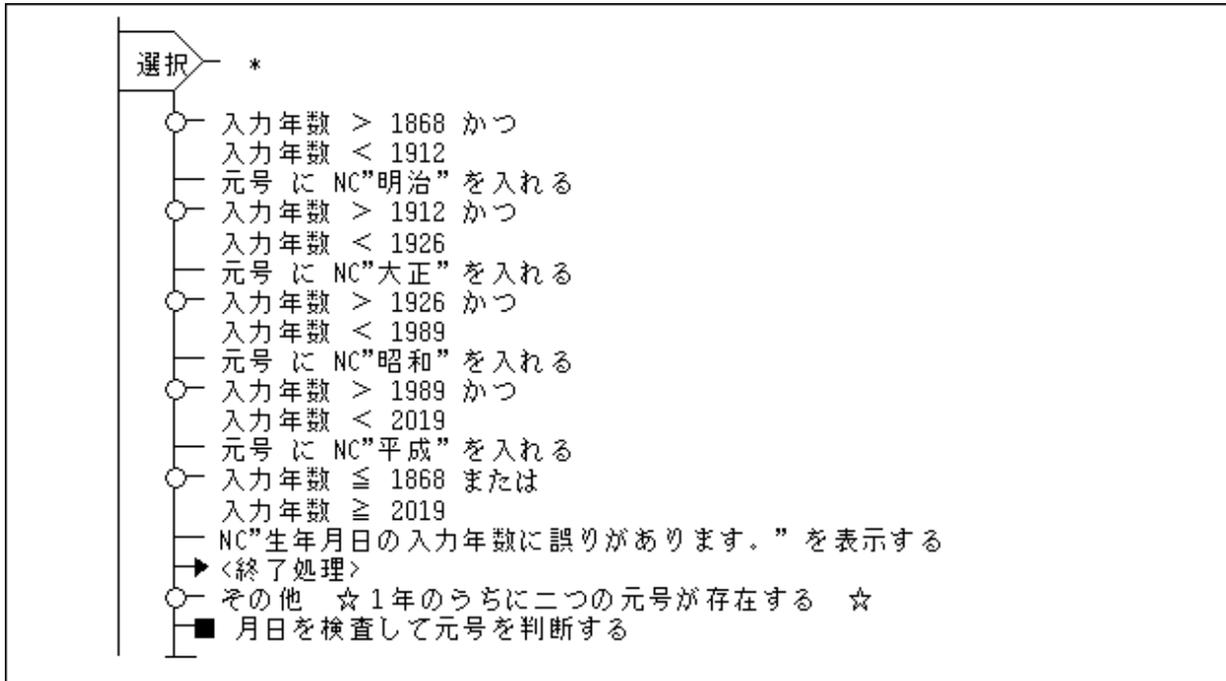
条件指定節に、手続き部の節を構成する表記を1つも記述しなかった場合には、空文が省略されているものとみなされます。

処理の流れ



備考:「処理」は、条件指定節に記述された処理を表しています。

記述例



3.10.2.4 複合選択(定数指定)

複合選択(定数指定)は、変数と定数を上から比較して、値が一致した条件指定節の処理をすべて実行します。

記述形式



記述規則

複選図記号の本文には、変数を記述します。

複合選択(定数指定)には、1つ以上の条件指定節を記述します。

変数

変数には、COBOLの一意名を記述します。

条件指定節

条件指定節は、変数と定数の値が一致したときの処理を記述するところです。

条件指定節の開始を表す丸図記号の本文には、定数を記述します。

条件指定節には、手続き部の節を構成する表記を記述します。

定数

定数には、COBOLの定数を記述します。一意名と定数の項類は比較可能な組合せでなければなりません。

一般規則

条件指定節に、手続き部の節を構成する表記を1つも記述しなかった場合には、空文が省略されているものとみなされます。



参照

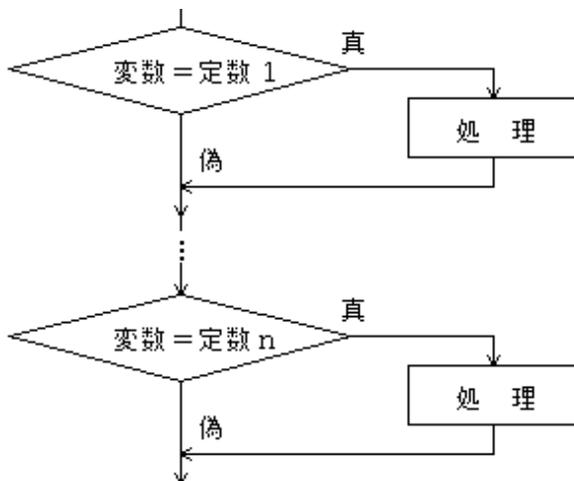
COBOLの用語

一意名

定数

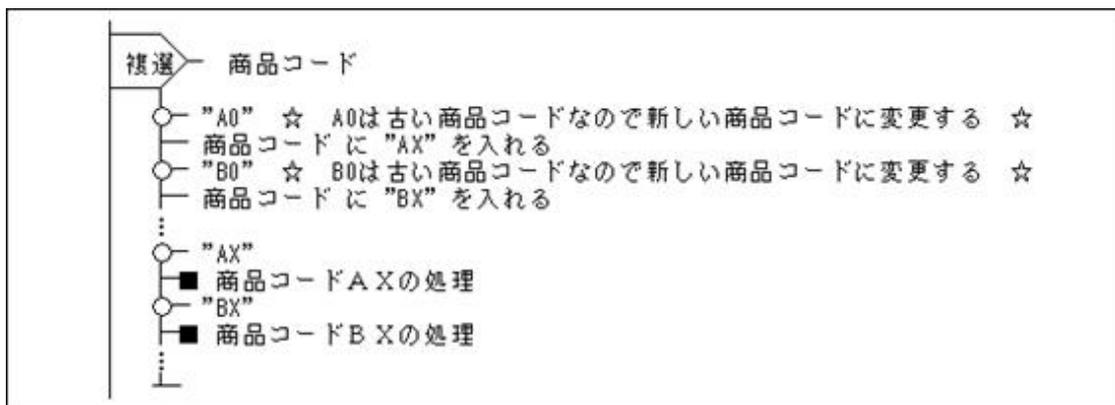
項類

処理の流れ



備考:「処理」は、条件指定節に記述された処理を表しています。

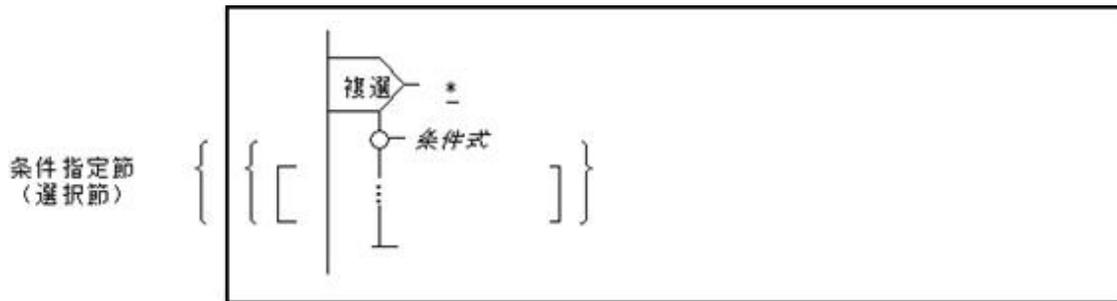
記述例



3.10.2.5 複合選択(条件式指定)

複合選択(条件式指定)は、条件式を上から評価し、条件式が真になった条件指定節の処理をすべて実行します。

記述形式



記述規則

複選図記号の本文には、システム定義のキーワード「*」を記述します。「*」は全角でも半角でも記述できます。

複合選択(条件式指定)には、1つ以上の条件指定節を記述します。

条件指定節

条件指定節は、条件式が真になったときの処理を記述するところです。

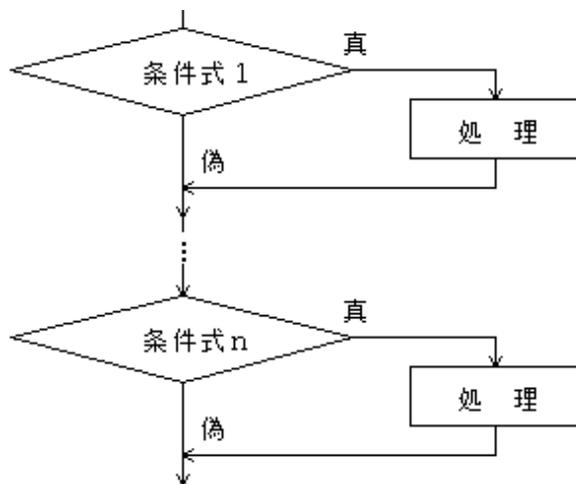
条件指定節の開始を表す丸図記号の本文には、条件式を記述します。記述規則については“[7.3.3 条件式](#)”を参照してください。

条件指定節には、手続き部の節を構成する表記を記述します。

一般規則

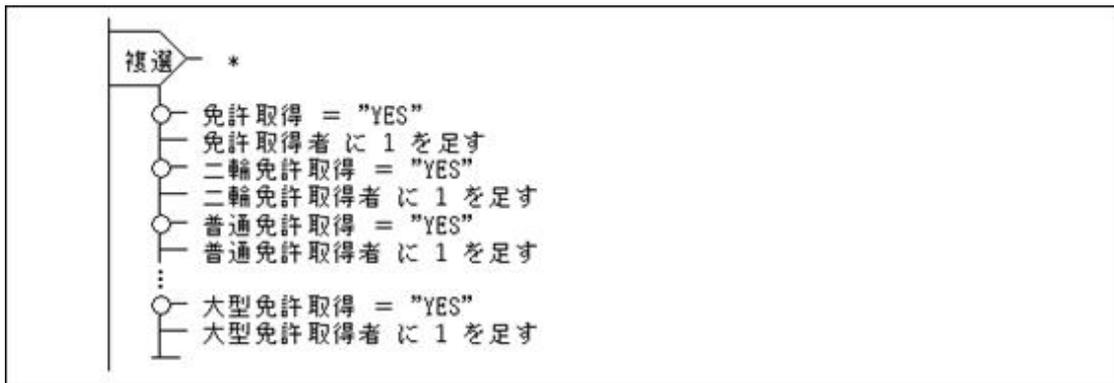
条件指定節に、手続き部の節を構成する表記を1つも記述しなかった場合には、空文が省略されているものとみなされます。

処理の流れ



備考:「処理」は、条件指定節に記述された処理を表しています。

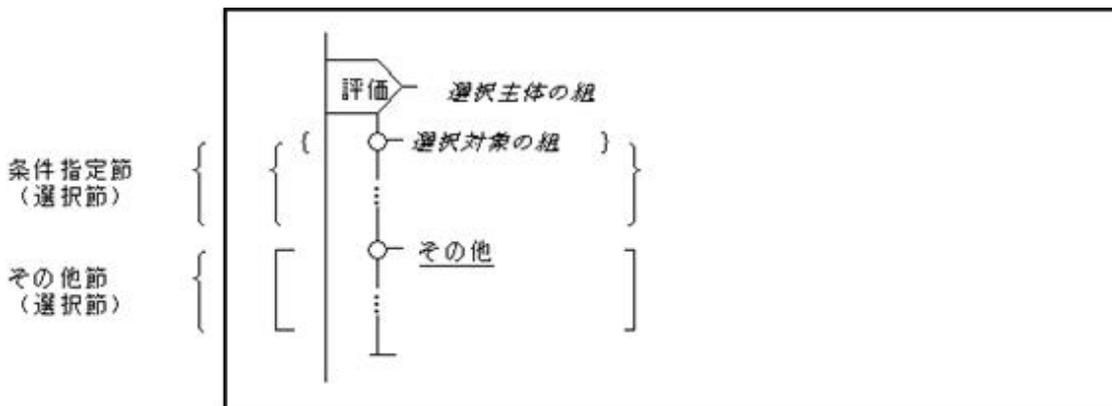
記述例



3.10.2.6 評価

評価は、選択主体の組と選択対象の組を評価し、評価結果が真である条件指定節の処理を実行します。COBOLのEVALUATE文に対応する表記です。

記述形式



記述規則

評価図記号の本文には、選択主体の組を記述します。

評価には、1つ以上の条件指定節とその他節を記述形式で示した順序で記述します。その他節は省略することができます。

選択主体の組

選択主体の組には、COBOLの選択主体の組を記述します。

条件指定節

条件指定節は、選択主体の組と選択対象の組の評価結果が真になったときの処理を記述するところです。

条件指定節の先頭には、丸図記号の本文に選択対象の組を記述した表記を1つ以上連続して記述します。

条件指定節には、手続き部の節を構成する表記を1つ以上記述します。

選択対象の組

選択対象の組には、COBOLの選択対象の組を記述します。

その他節

その他節は、選択主体の組と選択対象の組の評価結果が真になる条件指定節がなかったときの処理を記述するところです。

その他節の開始を表す丸図記号の本文には、システム定義のキーワード「その他」、「OTHER」または「other」を記述します。「OTHER」および「other」は全角でも半角でも記述できます。

その他節には、手続き部の節を構成する表記を1つ以上記述します。

参照

COBOLの用語

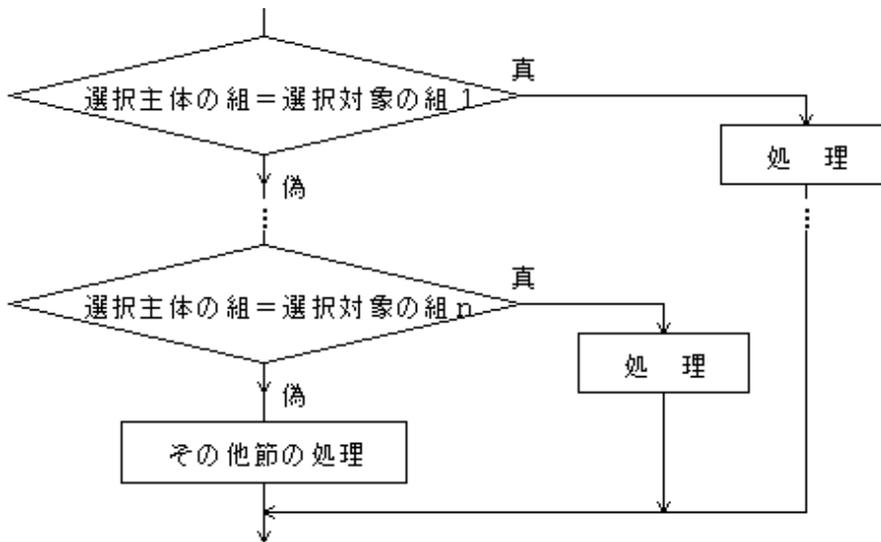
EVLUATE文

選択主体の組 ⇒ EVALUATE文

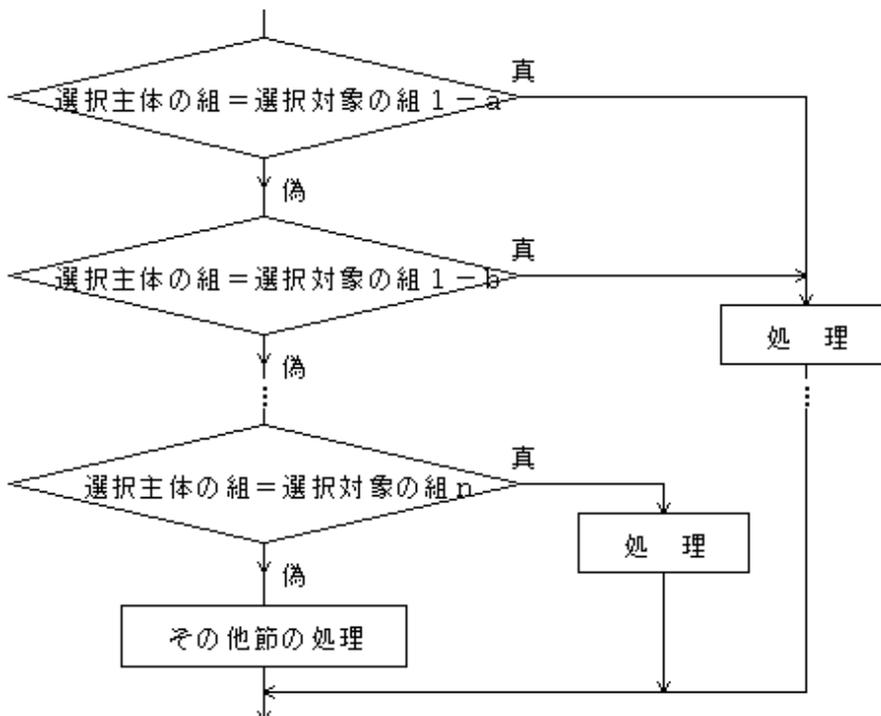
選択対象の組 ⇒ EVALUATE文

処理の流れ

[すべての条件指定節に丸図記号が1つだけ記述されている場合]

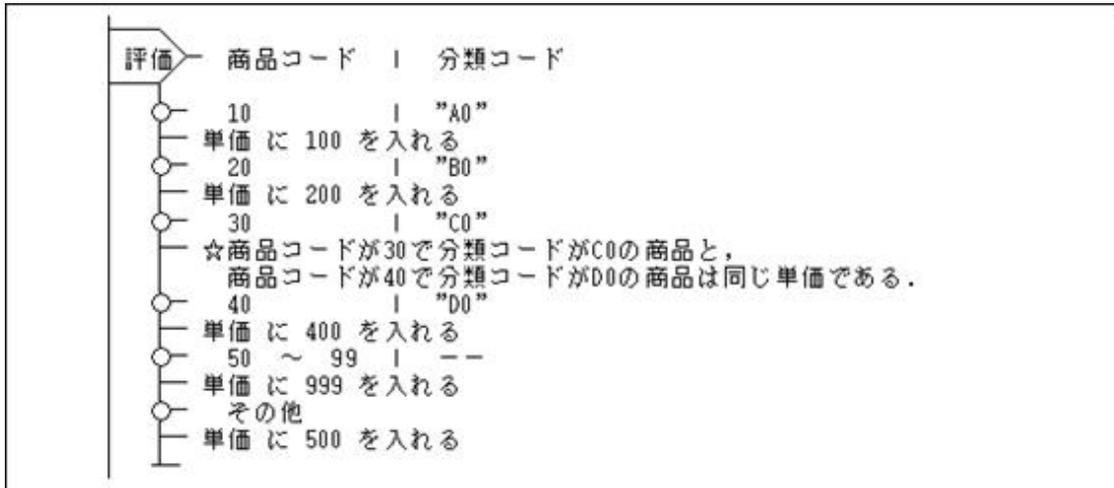


[最初の条件指定節に丸図記号が2つ記述されている場合]



備考:「処理」は、条件指定節に記述された処理を表しています。

記述例



3.10.3 ループ系の表記

ループ系の表記には、以下の12種類があります。

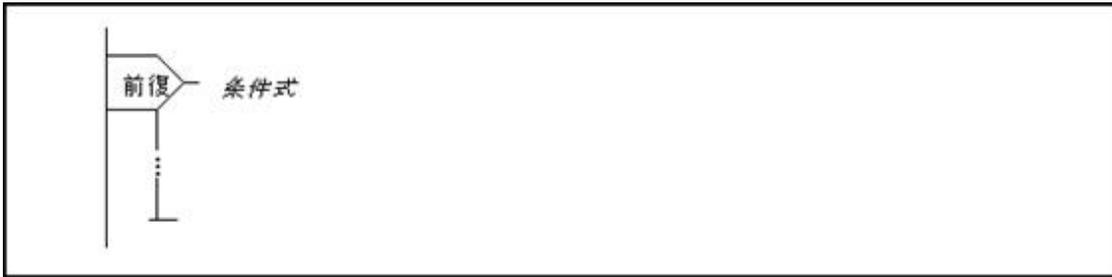
- ・ 前判定ループ
- ・ 後判定ループ
- ・ 終了条件前判定ループ
- ・ 変数型ループ
- ・ 無限ループ
- ・ 指定回数ループ
- ・ 更新型ループ
- ・ 変数型ループと前判定ループの組合せ
- ・ 変数型ループと後判定ループの組合せ
- ・ 変数型ループと終了条件前判定ループの組合せ
- ・ 更新型ループと前判定ループの組合せ
- ・ 更新型ループと後判定ループの組合せ

3.10.3.1 前判定ループ

前判定ループは、条件式が真の間、従属する処理を繰り返し実行します。

条件式の評価は、従属する処理を実行する前に行われます。したがって、初めから条件式が偽の場合には、従属する処理は一度も実行されません。

記述形式

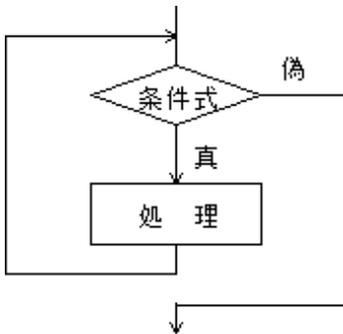


記述規則

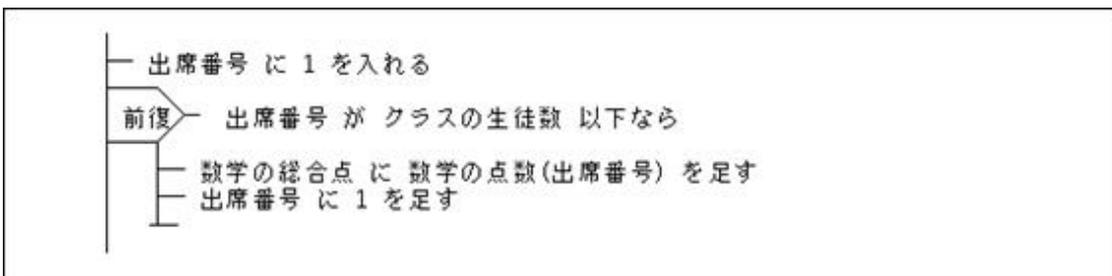
前復図記号の本文には、条件式を記述します。条件式には、処理を継続する条件を記述します。記述規則については“7.3.3 条件式”を参照してください。

前判定ループには、手続き部の節を構成する表記を1つ以上記述します。

処理の流れ



記述例



3.10.3.2 後判定ループ

後判定ループは、条件式が真になるまで、従属する処理を繰り返し実行します。

条件式の評価は、従属する処理を実行した後に行われます。したがって、初めから条件式が真の場合にも、従属する処理は、一度は実行されます。

記述形式

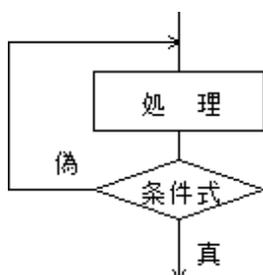


記述規則

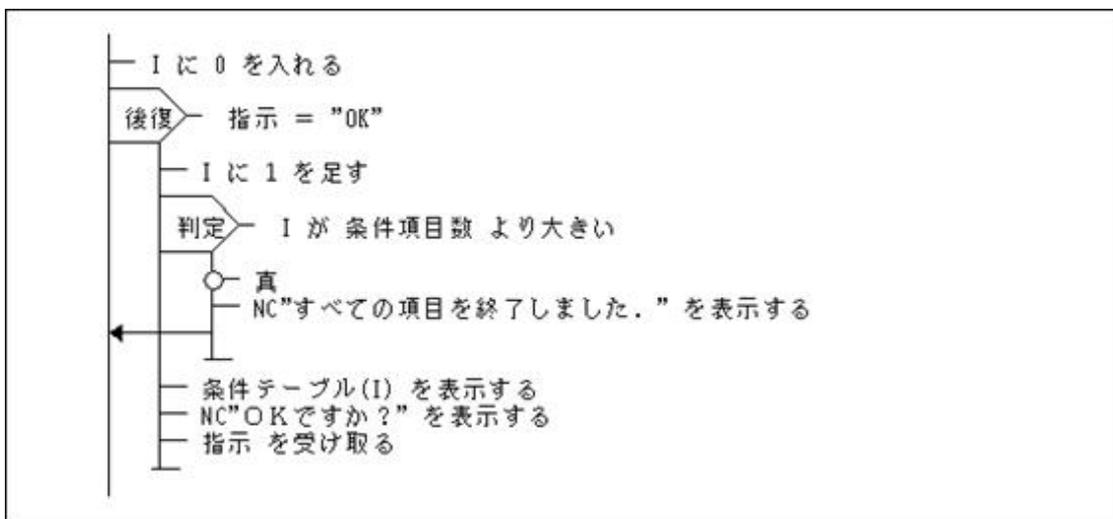
後復図記号の本文には、条件式を記述します。条件式には、処理を終了する条件を記述します。記述規則については“7.3.3 条件式”を参照してください。

後判定ループには、手続き部の節を構成する表記を1つ以上記述します。

処理の流れ



記述例



3.10.3.3 終了条件前判定ループ

終了条件前判定ループは、条件式が真になるまで、従属する処理を繰り返し実行します。

条件式の評価は、従属する処理を実行する前に行われます。したがって、初めから条件式が真の場合には、従属する処理は一度も実行されません。

記述形式

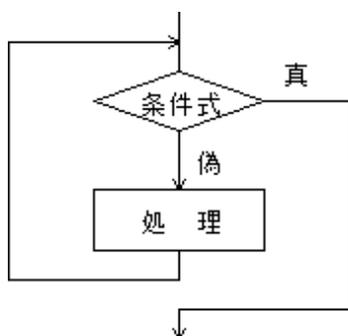


記述規則

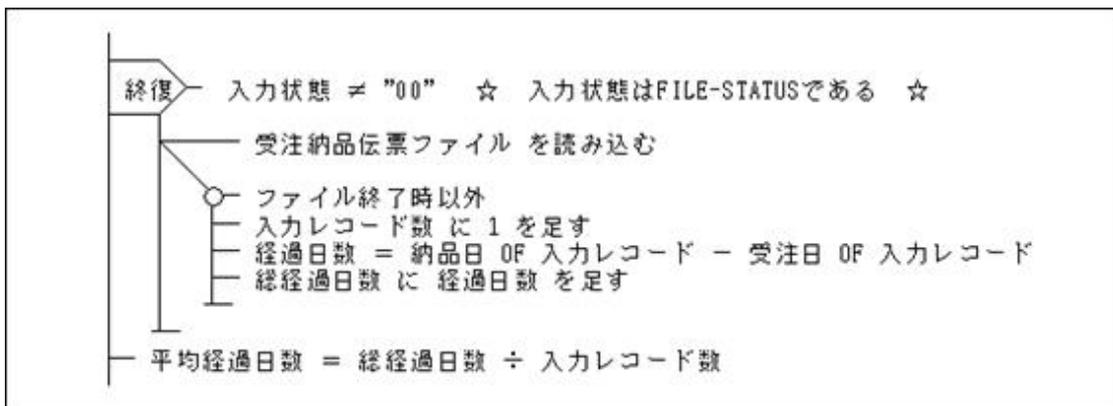
終復図記号の本文には、条件式を記述します。条件式には、処理を終了するための条件を記述します。記述規則については“7.3.3 条件式”を参照してください。

終了条件前判定ループには、手続き部の節を構成する表記を1つ以上記述します。

処理の流れ



記述例



3.10.3.4 変数型ループ

変数型ループは、変数が最終値より大きく(または小さく)なるまで、従属する処理を繰り返し実行します。

初めに変数に初期値を設定した後、変数と最終値を比較し、変数が最終値より大きく(小さく)なければ、従属する処理を実行します。2回目以降は、変数に増分値を加算(減算)した後、変数と最終値を比較し、変数が最終値より大きく(小さく)なければ、従属する処理を実行します。

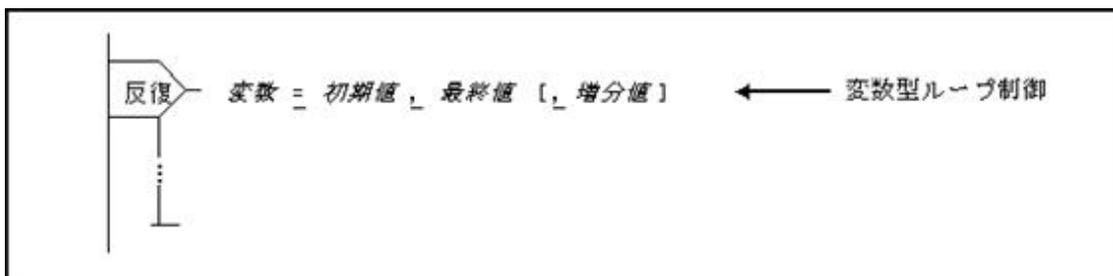
初めから変数が最終値より大きい(または小さい)場合には、従属する処理は一度も実行されません。

記述形式

[形式1]



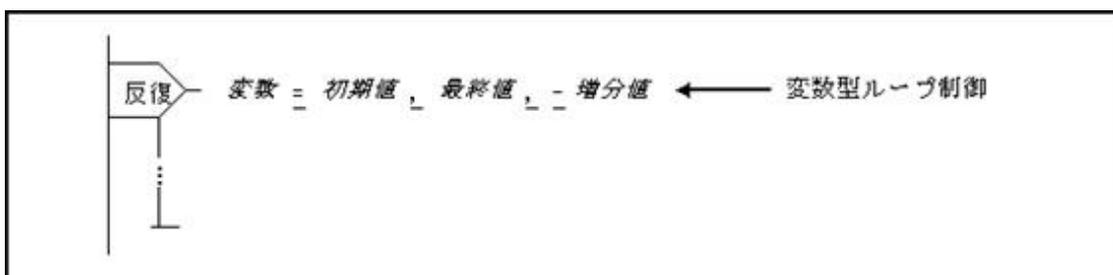
[形式2]



[形式3]



[形式4]



記述規則

反復図記号の本文には、変数型ループ制御を記述します。

変数型ループには、手続き部の節を構成する表記を1つ以上記述します。

変数型ループ制御

変数型ループ制御は、ループの繰り返し回数を制御するための式です。

[形式2]および[形式4]のシステム定義のキーワード「=」、「、」および「-」は、全角でも半角でも記述できます。

[形式1]および[形式2]で増分値が1の場合には、増分値を省略することができます。

変数

変数には、COBOLの一意名または指標名を記述します。

初期値

初期値には、COBOLの数字定数、またはCOBOLの項類が数字である一意名または指標名を記述します。

最終値

最終値には、COBOLの数字定数、またはCOBOLの項類が数字である一意名または指標名を記述します。

増分値

増分値には、COBOLの数字定数または項類が数字である一意名を記述します。



COBOLの用語

一意名

指標名

数字定数

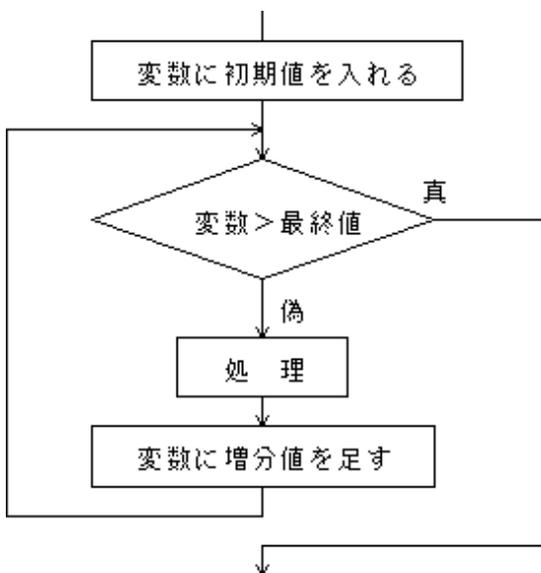
項類



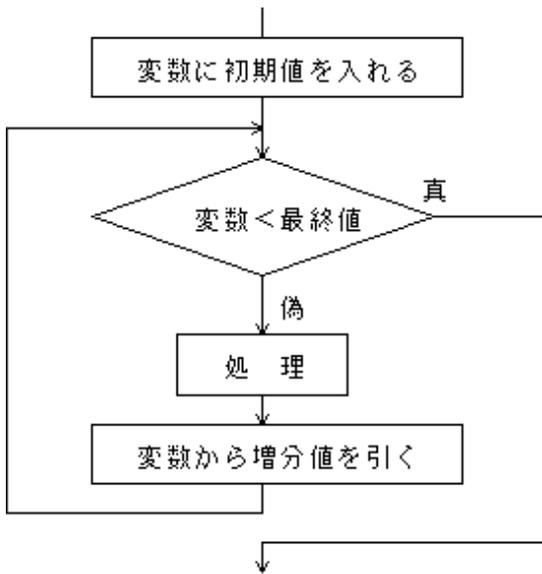
変数、最終値および増分値は、変数型ループに従属する処理の中で変更しないでください。

処理の流れ

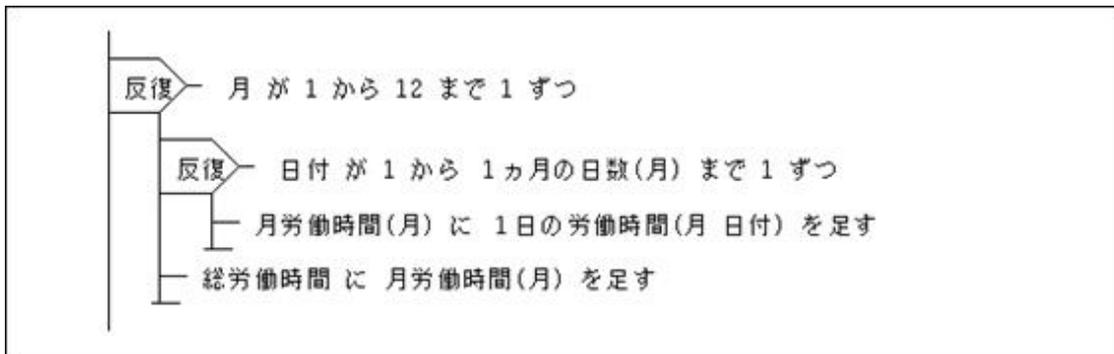
[形式1および形式2]



[形式3および形式4]



記述例



3.10.3.5 無限ループ

無限ループは、従属する処理を無限に繰り返し実行します。

記述形式



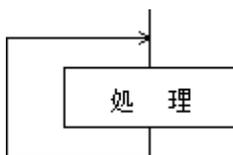
記述規則

無限ループは、手続き部の節内の主制御線上に記述します。

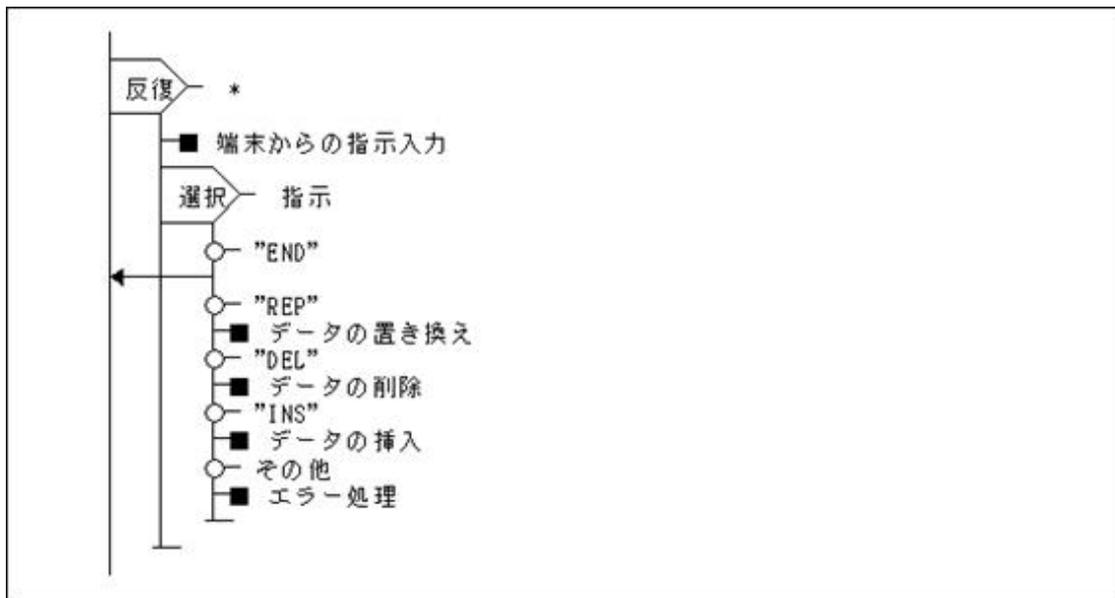
反復図記号の本文には、システム定義のキーワード「*」を記述します。「*」は全角でも半角でも記述できます。

無限ループには、手続き部の節を構成する表記を1つ以上記述します。表記にはループを抜けるための処理を含めてください。

処理の流れ



記述例



3.10.3.6 指定回数ループ

指定回数ループは、指定した回数だけ、従属する処理を繰り返し実行します。

指定した回数と実行した回数との比較は、従属する処理を実行する前に行われます。よって、指定した回数がゼロまたは負の場合には、従属する処理は一度も実行されません。

記述形式



記述規則

反復図記号の本文には、指定回数型ループ制御を記述します。

指定回数ループには、手続き部の節を構成する表記を1つ以上記述します。

指定回数型ループ制御

指定回数型ループ制御は、ループの繰り返し回数を制御するための式です。

指定回数型ループ制御には、定数または変数とシステム定義のキーワード「回」を記述します。

定数

定数には、COBOLの数字定数を記述します。

変数

変数には、COBOLの数字項目の一意名を記述します。



参照

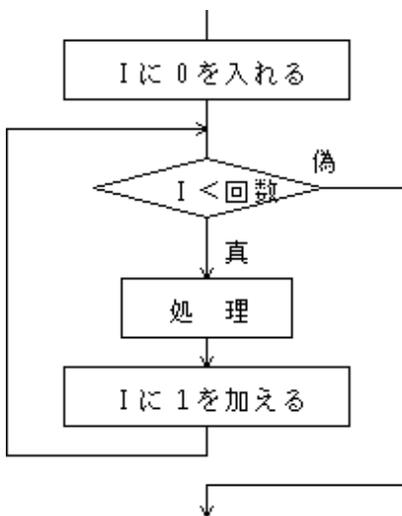
COBOLの用語

数字定数

数字項目

一意名

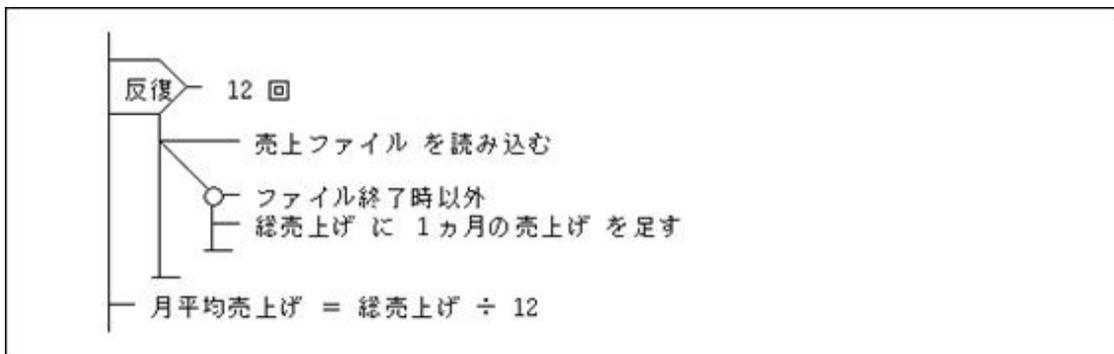
処理の流れ



備考: Iは作業用の変数を示しています。

処理を実行する回数は、従属する処理の最初の実行前に評価されます。よって、従属する処理の中で変数の値を変更しても、処理を実行する回数は変わりません。

記述例

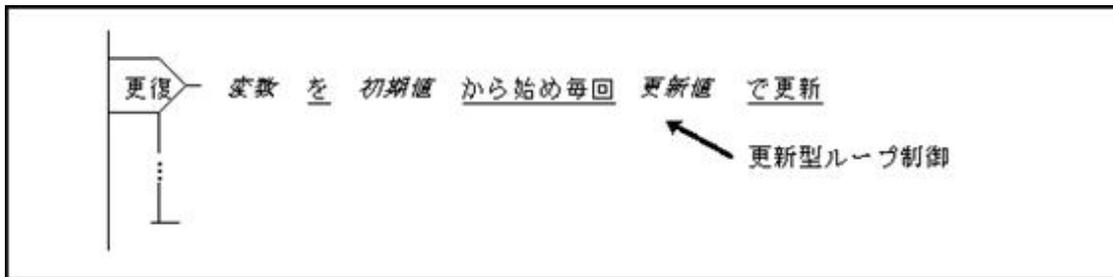


3.10.3.7 更新型ループ

更新型ループは、初めに変数に初期値を設定し、つづいて従属する処理を実行します。2回目以降は、変数に更新値を設定した後、従属する処理を実行します。この処理を無限に行います。

記述形式

[形式1]



[形式2]



記述規則

更新型ループは、手続き部の節内の主制御線上に記述します。

更復図記号の本文には、更新型ループ制御を記述します。

更新型ループには、手続き部の節を構成する表記を1つ以上記述します。表記にはループを抜けるための処理を含めてください。

更新型ループ制御

更新型ループ制御は、ループ内で使用する変数を制御するための式です。

[形式2]のシステム定義のキーワード「=」および「,」は、全角でも半角でも記述できます。

変数

変数には、COBOLの一意名を記述します。

初期値

初期値には、COBOLの定数または一意名を記述します。定数および一意名の項類は、変数に記述した一意名に転記可能な組合せでなければなりません。

更新値

更新値には、COBOLの定数または一意名を記述します。定数および一意名の項類は、変数に記述した一意名に転記可能な組合せでなければなりません。

参照

COBOLの用語

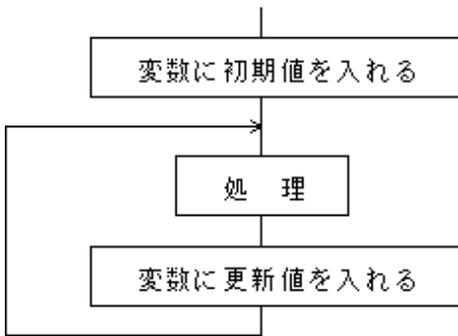
一意名

定数

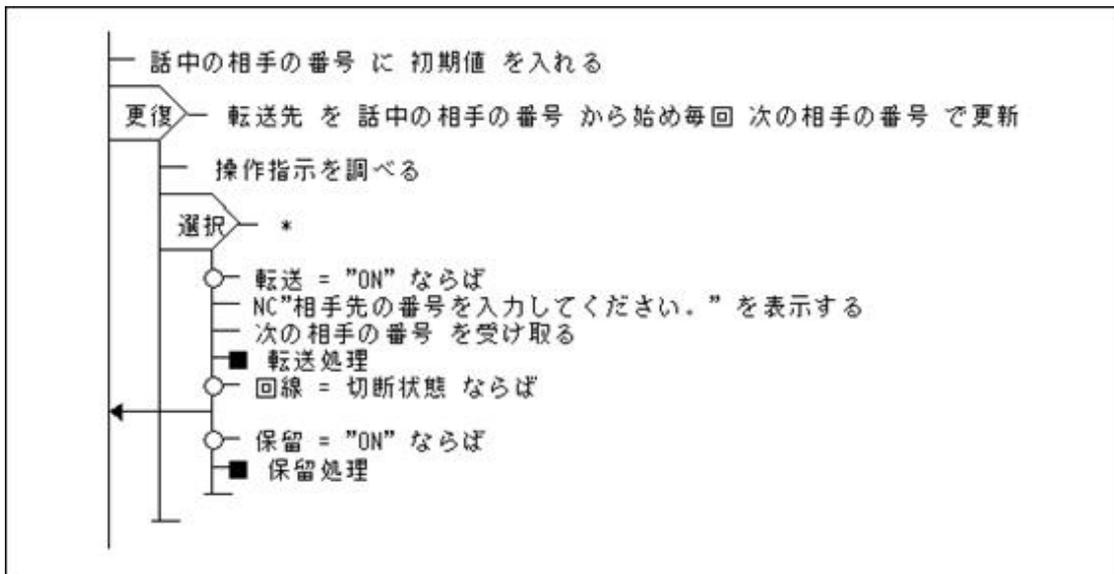
項類

転記

処理の流れ



記述例



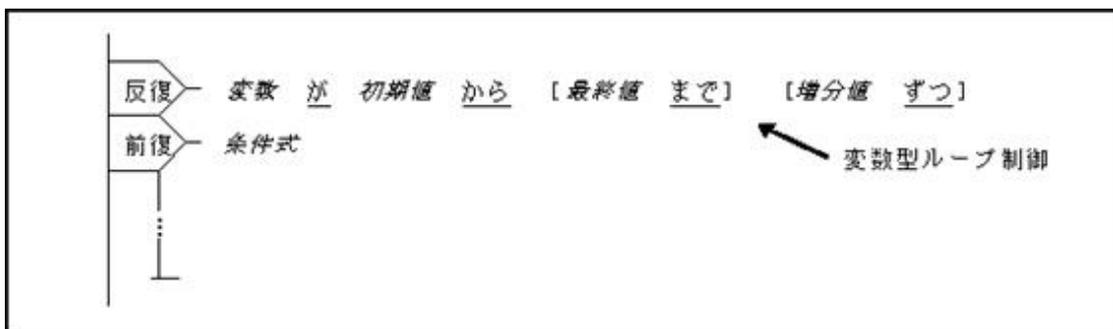
3.10.3.8 変数型ループと前判定ループの組合せ

変数型ループと前判定ループの組合せは、変数が最終値より大きく(または小さく)なるか、または条件式が偽になるまで、従属する処理を繰り返し実行します。

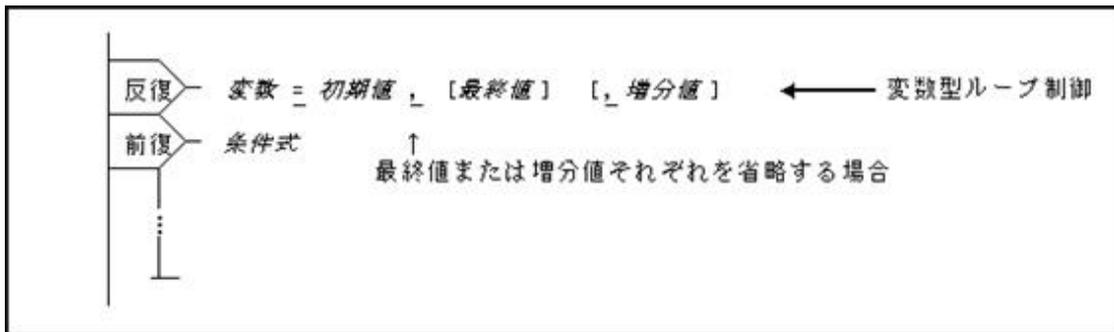
変数と最終値との比較および条件式の評価は、従属する処理を実行する前に行われます。よって、初めから変数が最終値より大きい(または小さい)場合、および初めから条件式が偽の場合には、従属する処理は一度も実行されません。

記述形式

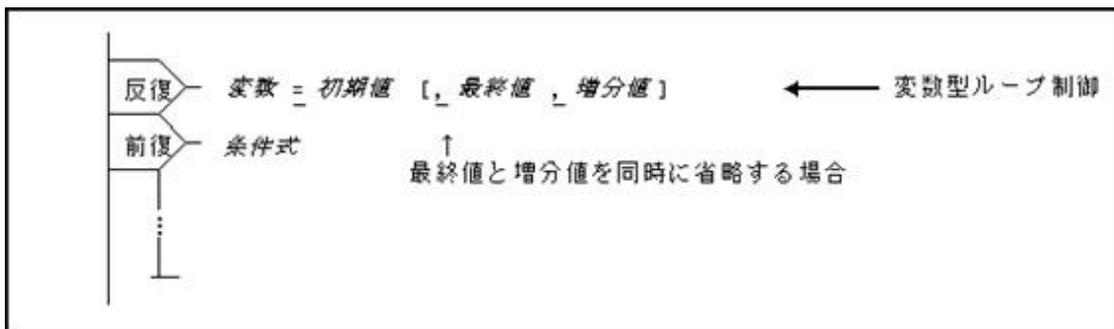
[形式1]



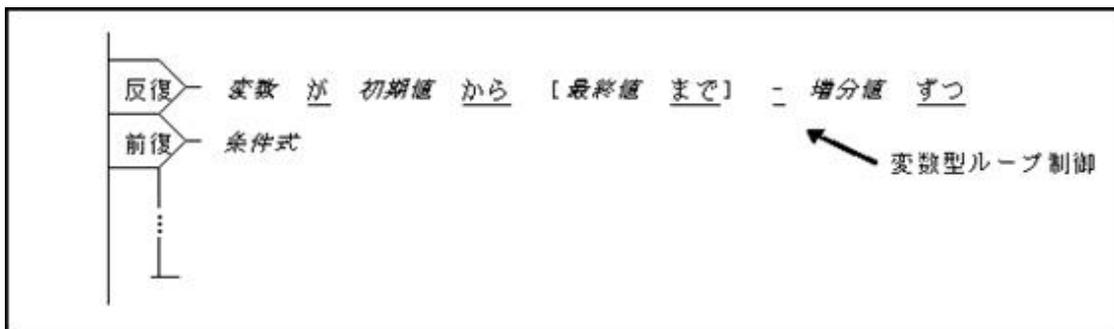
[形式2]



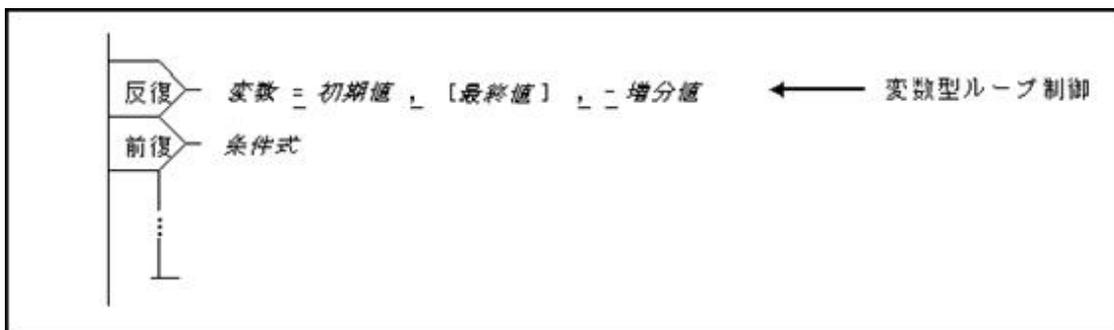
[形式3]



[形式4]



[形式5]



記述規則

反復図記号の本文には、変数型ループ制御を記述します。

前復図記号の本文には、条件式を記述します。条件式には、処理を継続する条件を記述します。記述規則については“7.3.3 条件式”を参照してください。

変数型ループと前判定ループの組合せには、手続き部の節を構成する表記を1つ以上記述します。

◆変数型ループ制御

変数型ループ制御は、ループの繰り返し回数を制御するための式です。

[形式2]、[形式3]および[形式5]のシステム定義のキーワード「=」、「」および「-」は、全角でも半角でも記述できます。

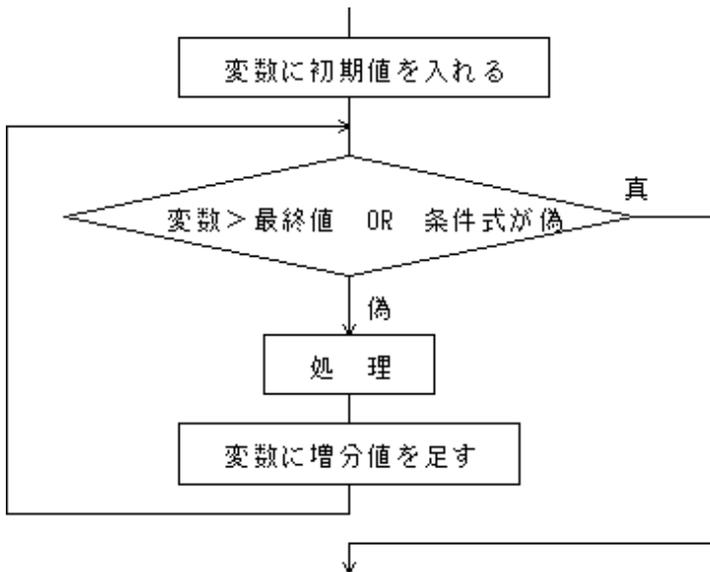
[形式1]、[形式2]および[形式3]で増分値が1の場合には、増分値を省略することができます。

変数型ループと前判定ループの組合せでは、最終値も省略することができます。

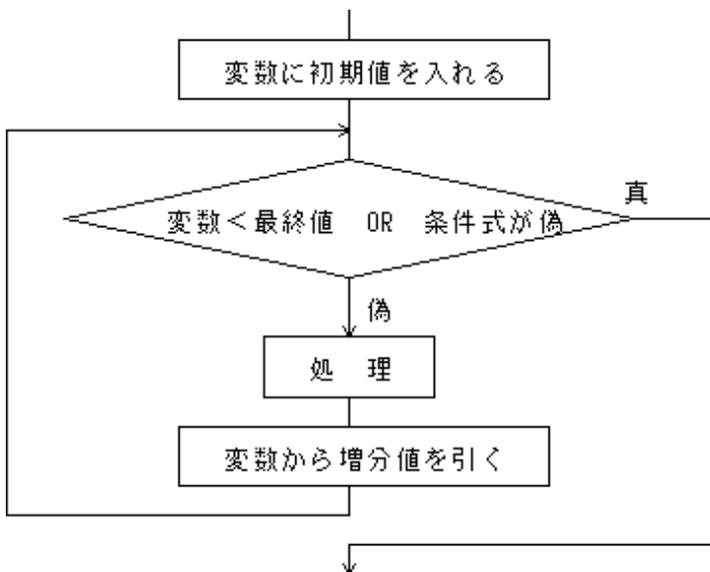
変数、初期値、最終値および増分値の記述規則については“[3.10.3.4 変数型ループ](#)”を参照してください。

処理の流れ

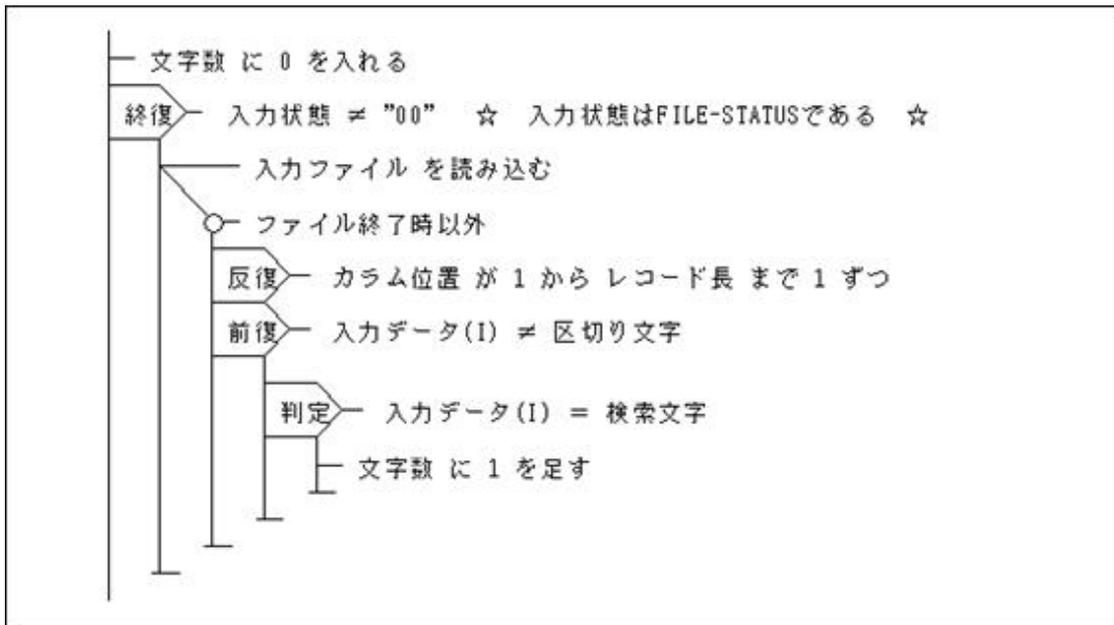
[形式1、形式2および形式3]



[形式4および形式5]



記述例



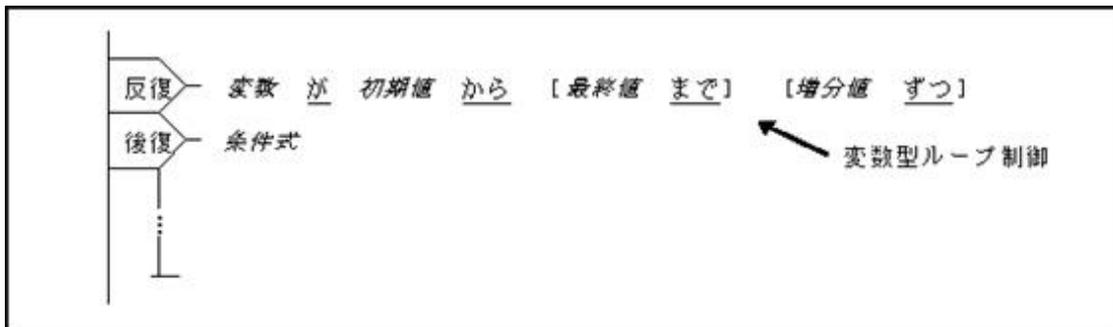
3.10.3.9 変数型ループと後判定ループの組合せ

変数型ループと後判定ループの組合せは、変数が最終値より大きく(または小さく)なるか、または条件式が真になるまで、従属する処理を繰り返し実行します。

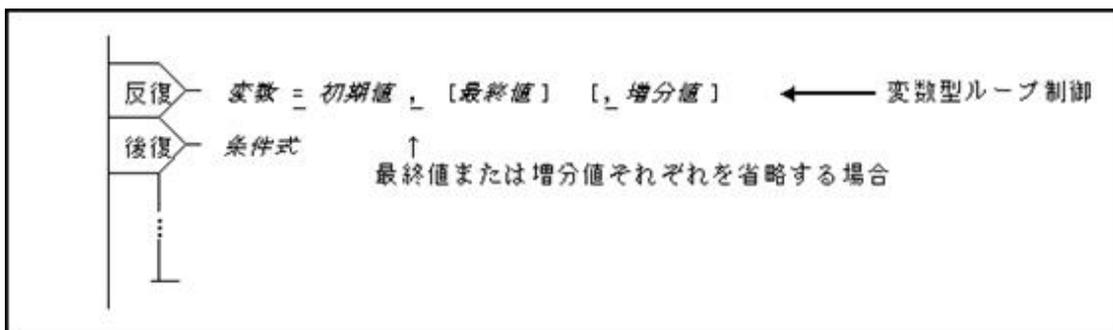
変数と最終値との比較は従属する処理を実行する前に、条件式の評価は従属する処理を実行した後に行われます。よって、初めから変数が最終値より大きい(または小さい)場合には、従属する処理は一度も実行されません。

記述形式

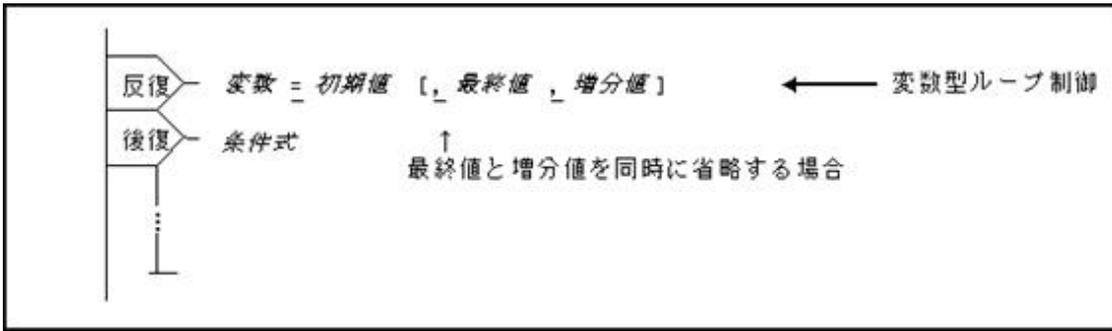
[形式1]



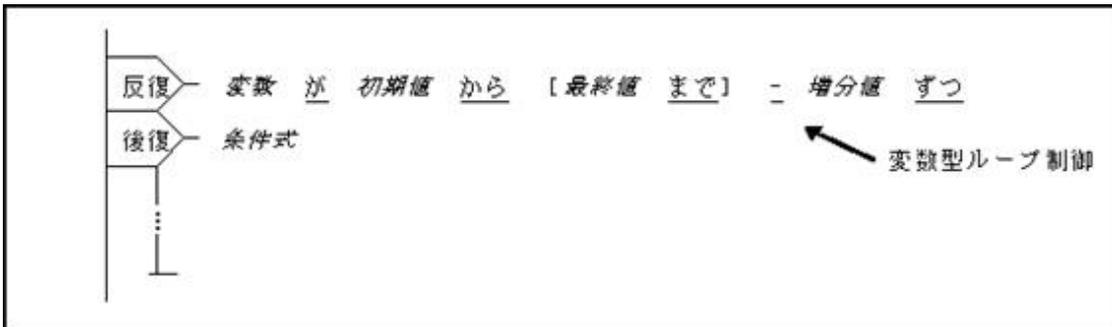
[形式2]



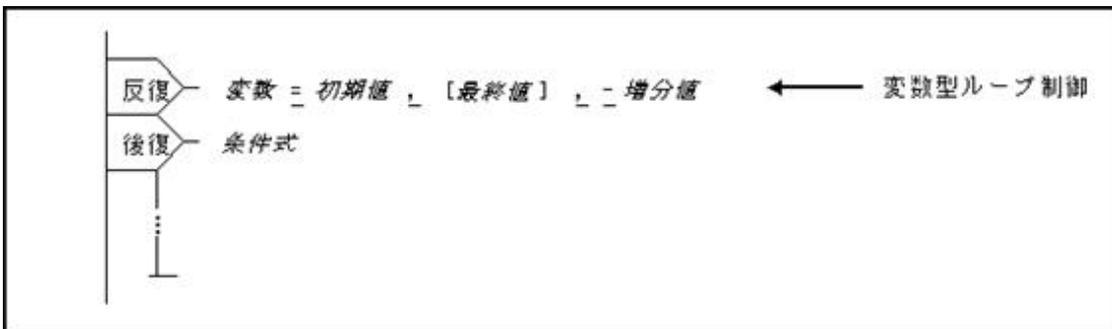
[形式3]



[形式4]



[形式5]



記述規則

反復図記号の本文には、変数型ループ制御を記述します。

後復図記号の本文には、条件式を記述します。条件式には、処理を終了する条件を記述します。記述規則については“[7.3.3 条件式](#)”を参照してください。

変数型ループと後判定ループの組合せには、手続き部の節を構成する表記を1つ以上記述します。

変数型ループ制御

変数型ループ制御は、ループの繰り返し回数を制御するための式です。

[形式2]、[形式3]および[形式5]のシステム定義のキーワード「=」、「、」および「-」は、全角でも半角でも記述できます。

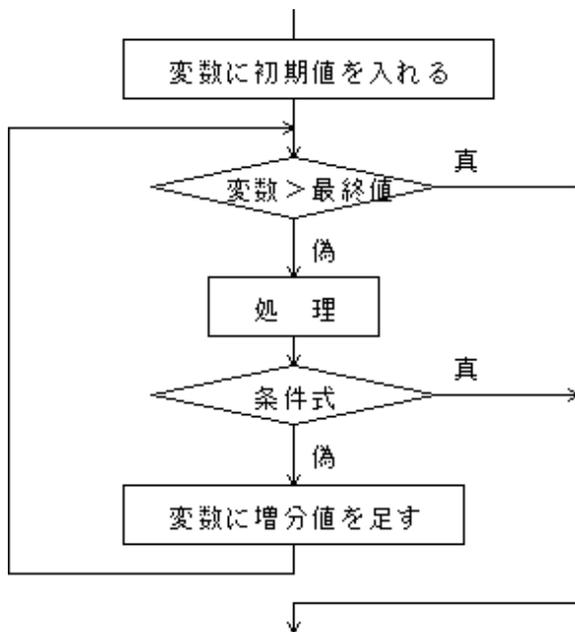
[形式1]、[形式2]および[形式3]で増分値が1の場合には、増分値を省略することができます。

変数型ループと後判定ループの組合せでは、最終値も省略することができます。

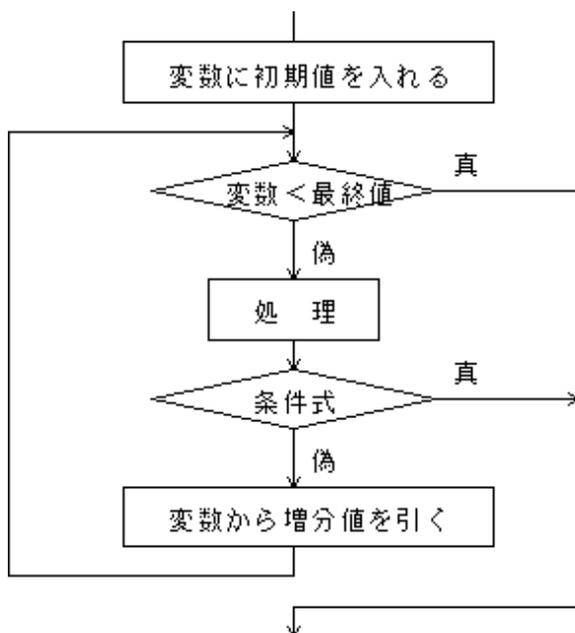
変数、初期値、最終値および増分値の記述規則については“[3.10.3.4 変数型ループ](#)”を参照してください。

処理の流れ

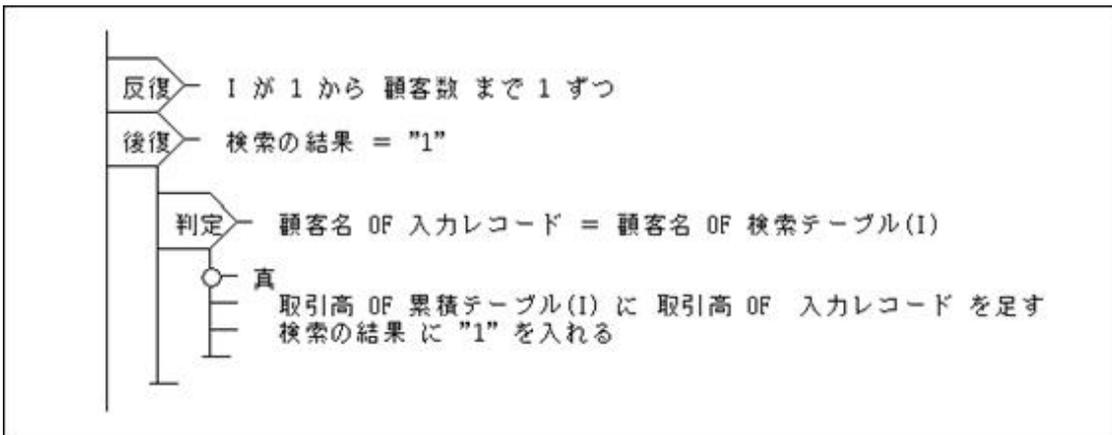
[形式1、形式2および形式3]



[形式4および形式5]



記述例



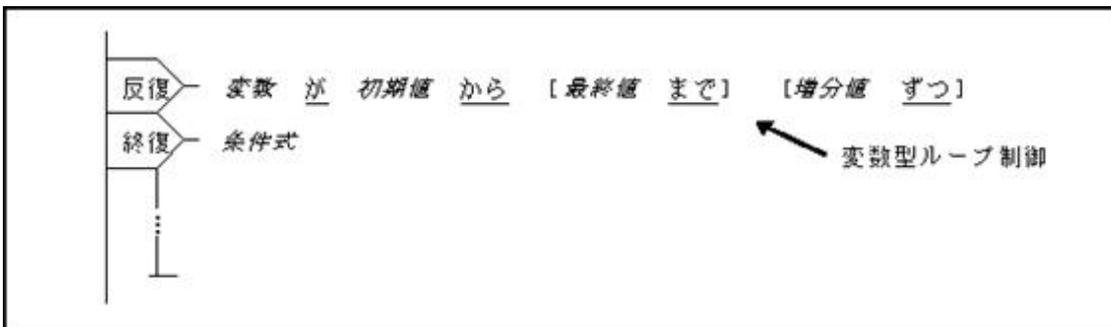
3.10.3.10 変数型ループと終了条件前判定ループの組合せ

変数型ループと終了条件前判定ループの組合せは、変数が最終値より大きく(または小さく)なるか、または条件式が真になるまで、従属する処理を繰り返し実行します。

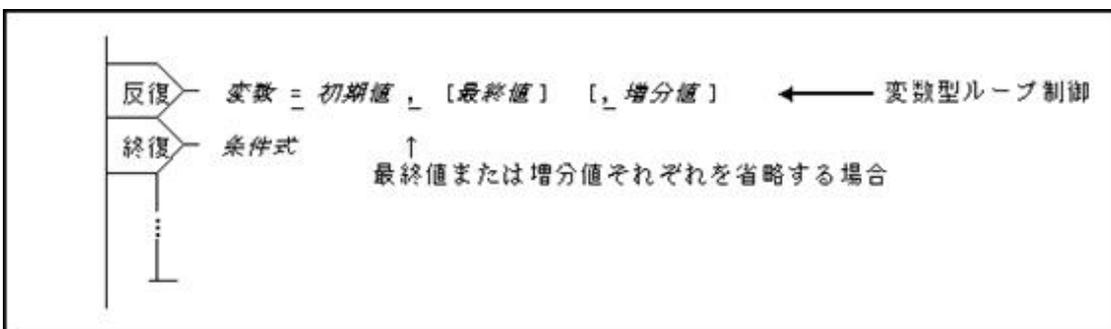
変数と最終値との比較および条件式の評価は、従属する処理を実行する前に行われます。よって、初めから変数が最終値より大きい(または小さい)場合、および初めから条件式が真の場合には、従属する処理は一度も実行されません。

記述形式

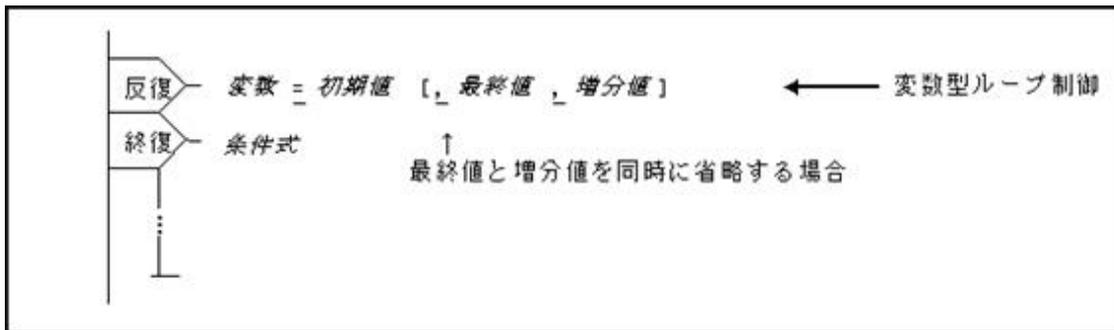
[形式1]



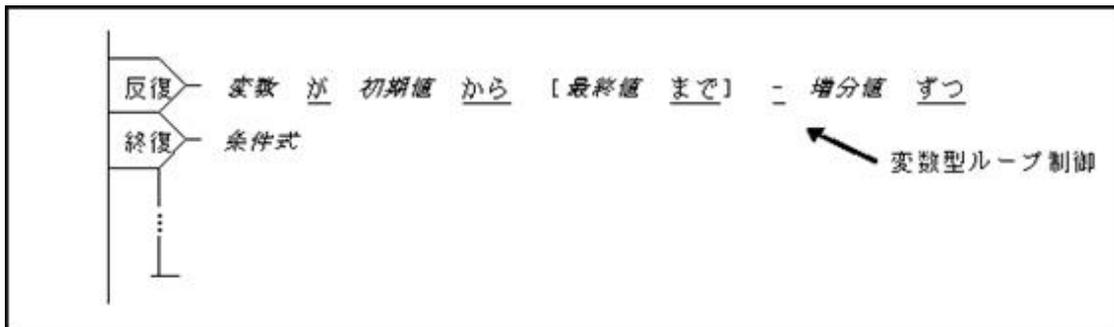
[形式2]



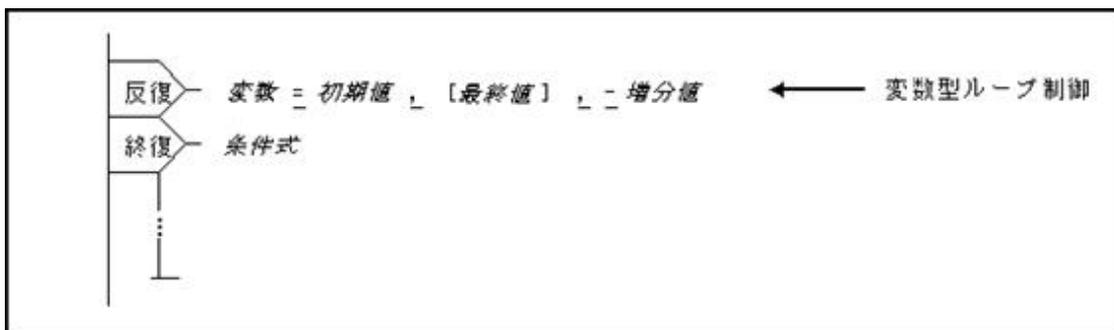
[形式3]



[形式4]



[形式5]



記述規則

反復図記号の本文には、変数型ループ制御を記述します。

終復図記号の本文には、条件式を記述します。条件式には、処理を終了する条件を記述します。記述規則については“[7.3.3 条件式](#)”を参照してください。

変数型ループと終了条件前判定ループの組合せには、手続き部の節を構成する表記を1つ以上記述します。

変数型ループ制御

変数型ループ制御は、ループの繰り返し回数を制御するための式です。

[形式2]、[形式3]および[形式5]のシステム定義のキーワード「=」、「」および「-」は、全角でも半角でも記述できます。

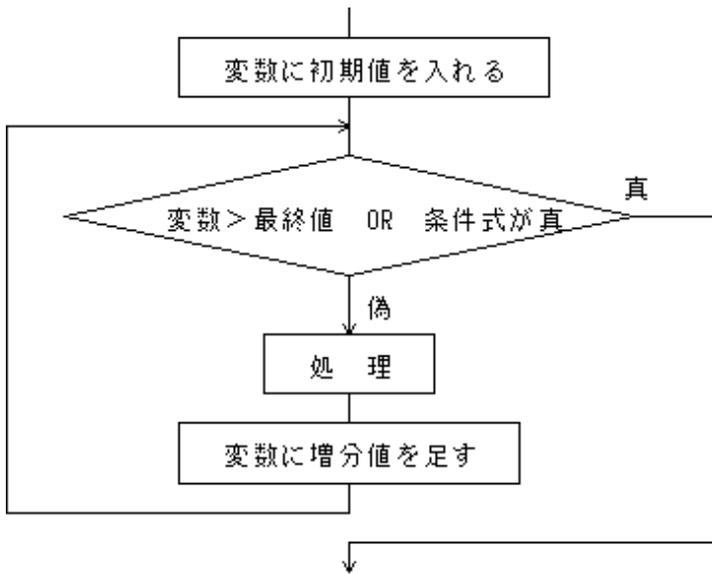
[形式1]、[形式2]および[形式3]で増分値が1の場合には、増分値を省略することができます。

変数型ループと終了条件前判定ループの組合せでは、最終値も省略することができます。

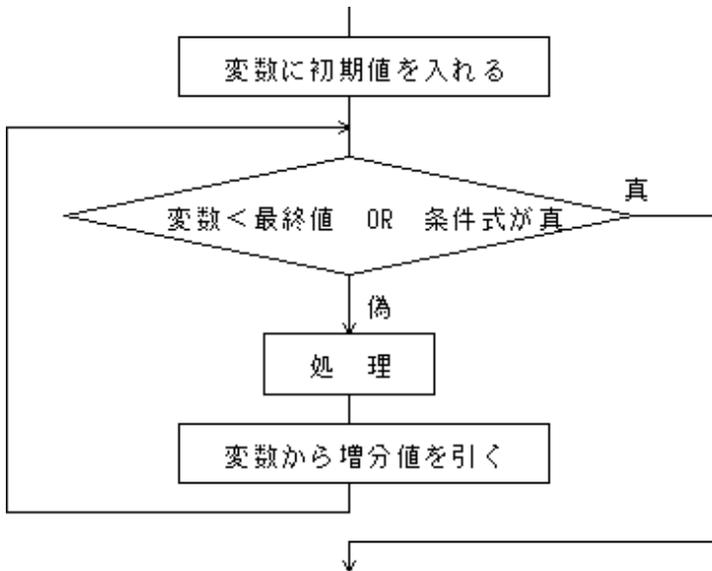
変数、初期値、最終値および増分値の記述規則については“[3.10.3.4 変数型ループ](#)”を参照してください。

処理の流れ

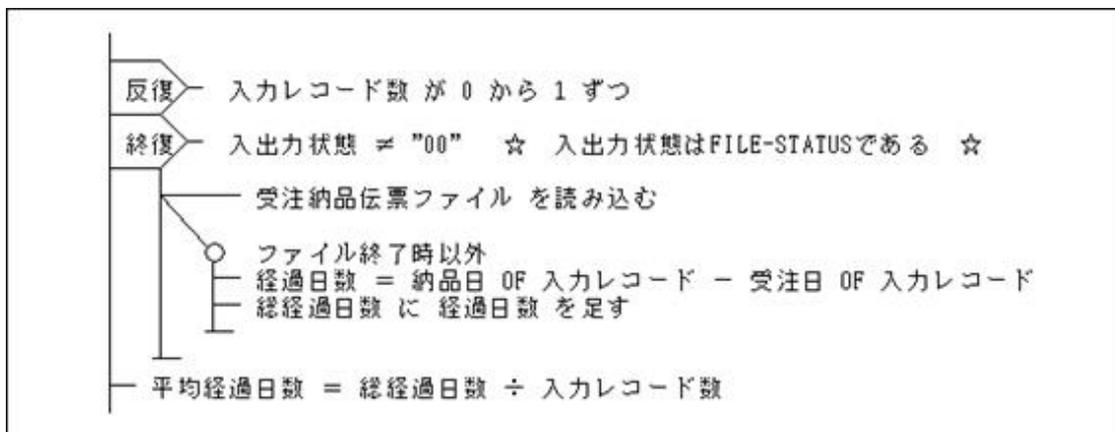
[形式1、形式2および形式3]



[形式4および形式5]



記述例



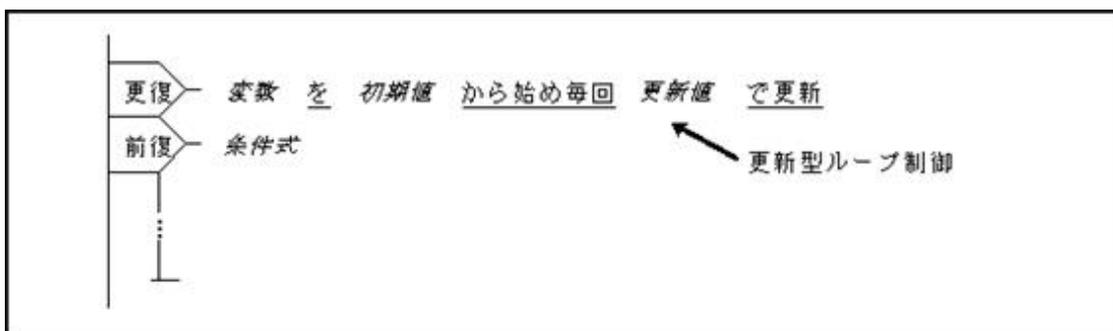
3.10.3.11 更新型ループと前判定ループの組合せ

更新型ループと前判定ループの組合せは、変数に初期値または更新値を設定した後、条件式が偽になるまで、従属する処理を繰り返し実行します。

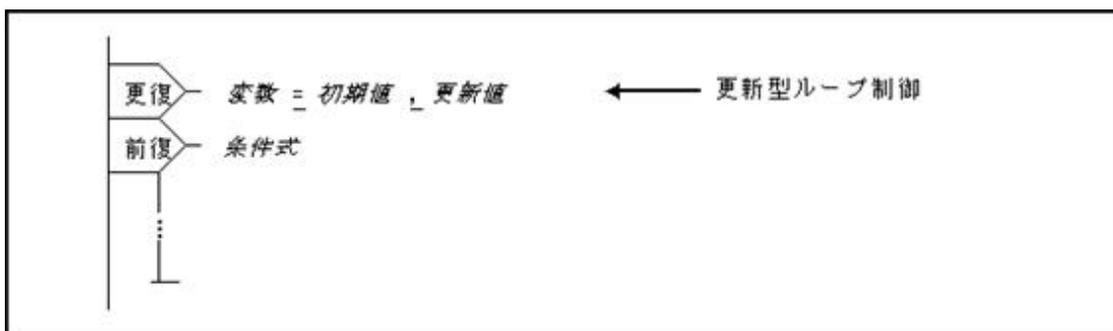
条件式の評価は、従属する処理を実行する前に行われます。よって、初めから条件式が偽の場合には、従属する処理は一度も実行されません。

記述形式

[形式1]



[形式2]



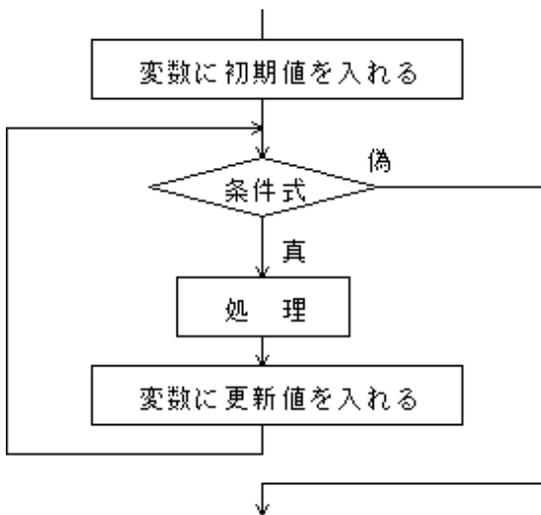
記述形式

更復図記号の本文には、更新型ループ制御を記述します。記述規則については“[3.10.3.7 更新型ループ](#)”を参照してください。

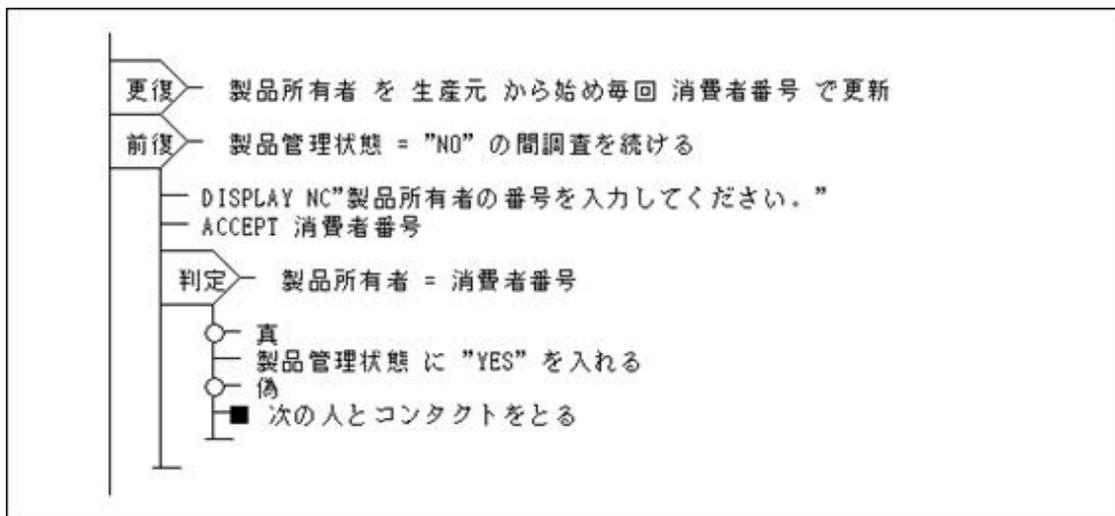
前復図記号の本文には、条件式を記述します。条件式には、処理を継続する条件を記述します。記述規則については“[7.3.3 条件式](#)”を参照してください。

更新型ループと前判定ループの組合せには、手続き部の節を構成する表記を1つ以上記述します。

処理の流れ



記述例



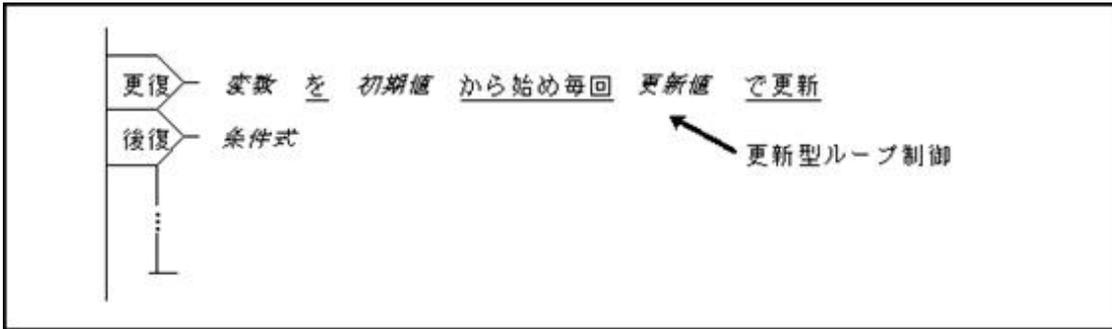
3.10.3.12 更新型ループと後判定ループの組合せ

更新型ループと後判定ループの組合せは、変数に初期値または更新値を設定した後、条件式が真になるまで、従属する処理を繰り返し実行します。

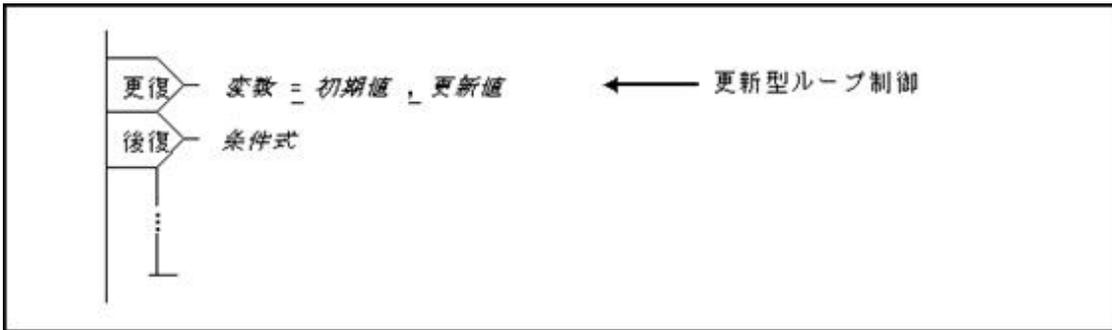
条件式の評価は、従属する処理を実行した後に行われます。よって、初めから条件式が真の場合にも、従属する処理は、一度は実行されます。

記述形式

[形式1]



[形式2]



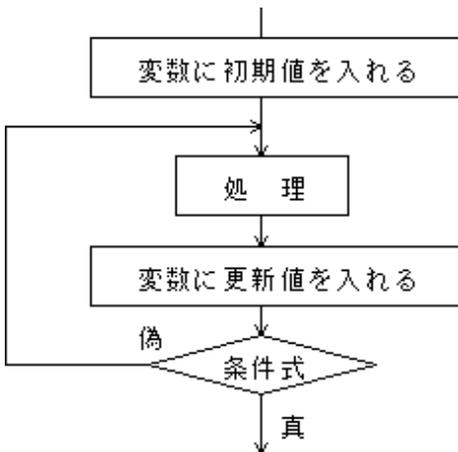
記述規則

更復図記号の本文には、更新型ループ制御を記述します。記述規則については“[3.10.3.7 更新型ループ](#)”を参照してください。

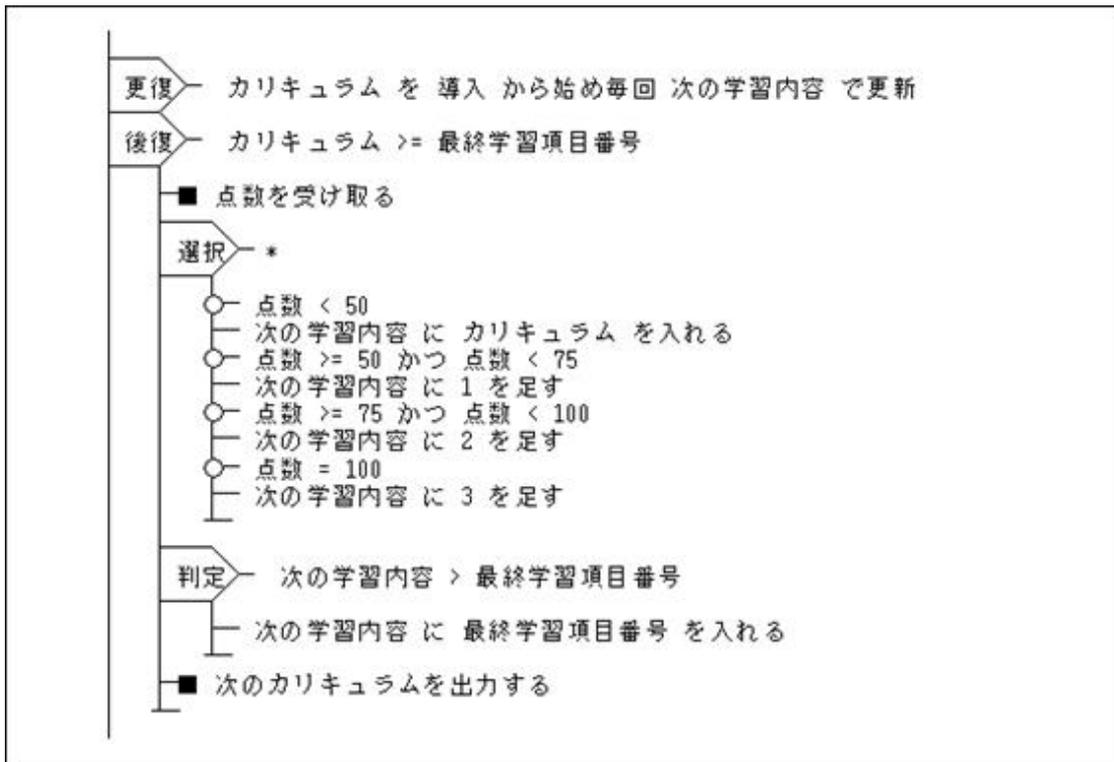
後復図記号の本文には、条件式を記述します。条件式には、処理を終了する条件を記述します。記述規則については“[7.3.3 条件式](#)”を参照してください。

更新型ループと後判定ループの組合せには、手続き部の節を構成する表記を1つ以上記述します。

処理の流れ



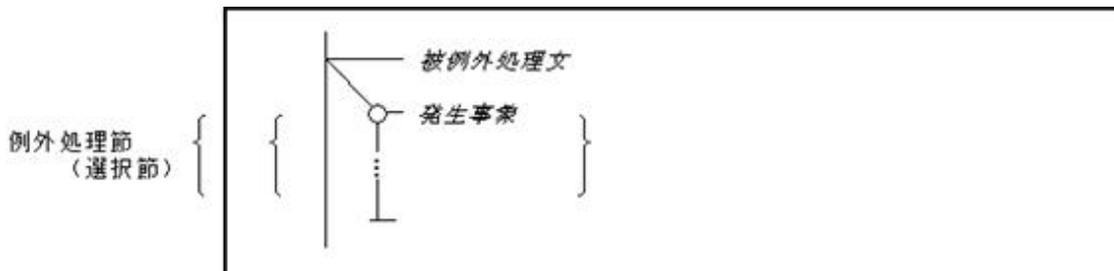
記述例



3.10.4 例外処理

例外処理は、ある処理を行って特定の事象が発生した場合に、その事象に対する処理を実行します。

記述形式



記述規則

例外処理図記号の本文には、被例外処理文を記述します。

例外処理には、1つ以上の例外処理節を記述します。

被例外処理文

被例外処理文には、COBOLの文を記述します。被例外処理文に記述できるCOBOLの文は限定されています。詳細は一般規則を参照してください。

例外処理節

例外処理節は、被例外処理文を実行したときに、特定の事象が発生した場合の処理を記述するところです。

例外処理節の開始を表す丸図記号の本文には、発生事象を記述します。発生事象については一般規則を参照してください。

例外処理節には、手続き部の節を構成する表記を1つ以上記述します。

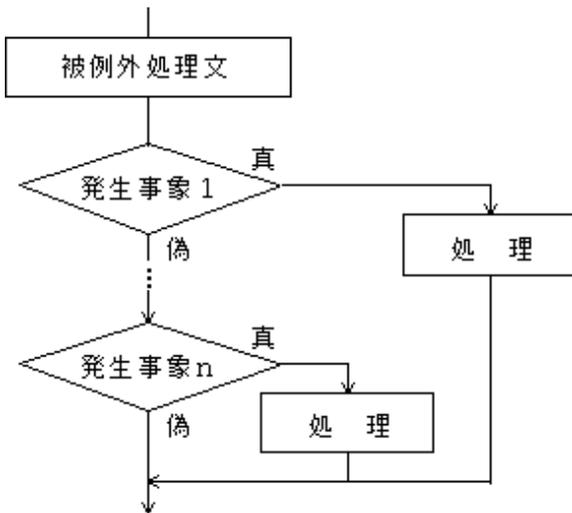
一般規則

被例外処理文に記述できるCOBOLの文と発生事象の組合せを“表3.2 被例外処理文と発生事象の組合せ”に示します。

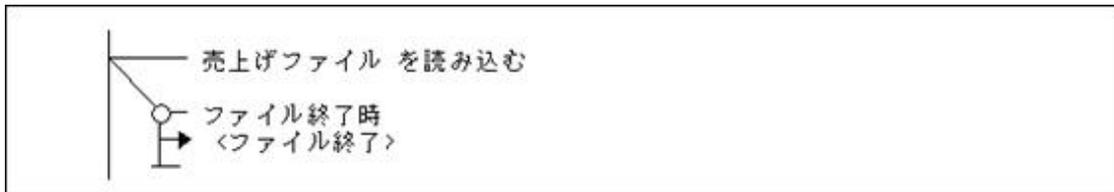
表3.2 被例外処理文と発生事象の組合せ

被例外処理文	発生事象
READ	[AT] END NOT [AT] END INVALID [KEY] NOT INVALID [KEY]
WRITE	INVALID [KEY] NOT INVALID [KEY] [AT] EOP NOT [AT] EOP [AT] END-OF-PAGE NOT [AT] END-OF-PAGE
DELETE REWRITE START	INVALID [KEY] NOT INVALID [KEY]
ADD COMPUTE DIVIDE MULTIPLY SUBTRACT	[ON] SIZE ERROR NOT [ON] SIZE ERROR
STRING UNSTRING	[ON] OVERFLOW NOT [ON] OVERFLOW
CALL	[ON] EXCEPTION NOT [ON] EXCEPTION [ON] OVERFLOW
FIND GET STORE	[AT] END
RETURN	[AT] END NOT [AT] END
[Win] ACCEPT DISPLAY	[ON] EXCEPTION NOT [ON] EXCEPTION

処理の流れ



記述例

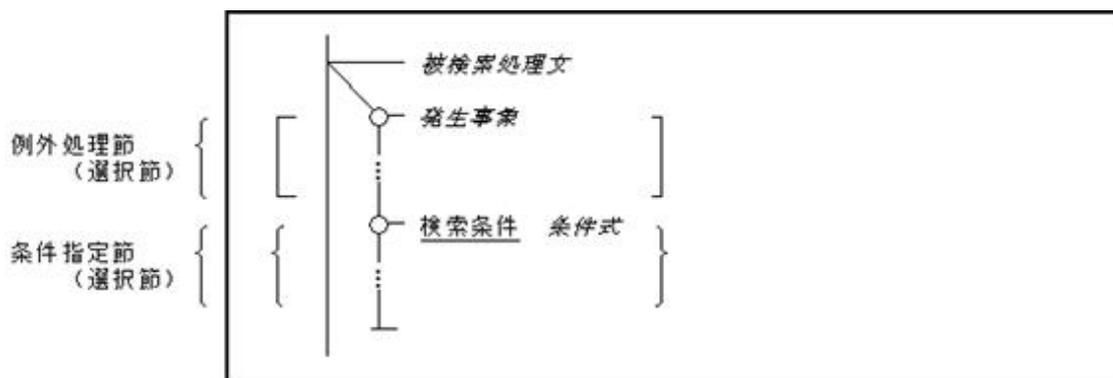


備考: 標準構文「～を読み込む」および標準日本語名標「ファイル終了時」を使用しています。

3.10.5 検索処理

検索処理は、指定した条件を満足する表要素を探し、対応する指標の値がその表要素を指すようにします。COBOLのSEARCH文に対応する表記です。

記述形式



記述規則

例外処理図記号の本文には、被検索処理文を記述します。

検索処理には、1つの例外処理節と1つ以上の条件指定節を記述形式で示した順序で記述します。例外処理節は省略することができます。

被検索処理文

被検索処理文には、COBOLの文を記述します。被検索処理文に記述できるCOBOLの文は限定されています。詳細は一般規則を参照してください。

例外処理節

例外処理節は、被検索処理文を実行したときに、特定の事象が発生した場合の処理を記述するところです。

例外処理節の開始を表す丸図記号の本文には、**発生事象**を記述します。発生事象については一般規則を参照してください。

例外処理節には、手続き部の節を構成する表記を1つ以上記述します。

条件指定節

条件指定節は、指定した条件を満足する表要素が見つかった場合の処理を記述するところです。

条件指定節の開始を表す丸図記号の本文には、システム定義のキーワード「検索条件」、「WHEN」または「when」と条件式を記述します。「WHEN」および「when」は全角でも半角でも記述できます。条件式については“7.3.3 条件式”を参照してください。

条件指定節には、手続き部の節を構成する表記を1つ以上記述します。



参照

COBOLの用語

表要素

指標

SEARCH文

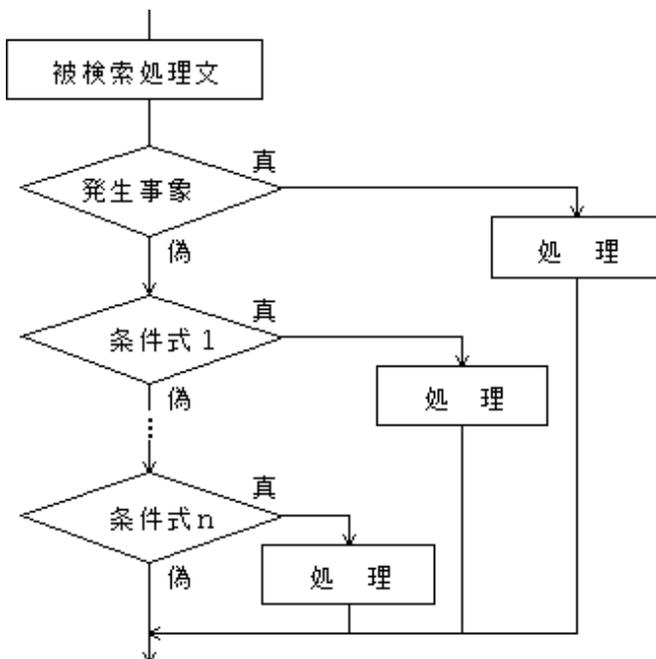
一般規則

被検索処理文に記述できるCOBOLの文と発生事象の組合せを“表3.3 被検索処理文と発生事象の組合せ”に示します。

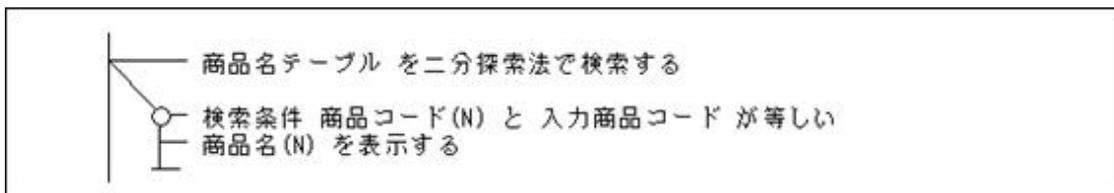
表3.3 被検索処理文と発生事象の組合せ

被検索処理文	発生事象
SEARCH	[AT] END

処理の流れ



記述例



備考: 標準構文「～を二分探索法で検索する」、「～と～が等しい」および「～を表示する」を使用しています。

3.10.6 制御変更系の表記

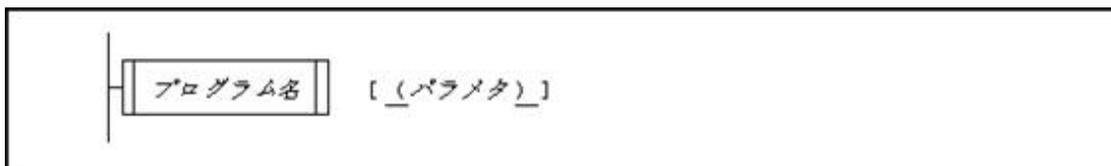
制御変更系の表記には、以下の10種類があります。

- ・ プログラム呼出し
- ・ 内部プログラム呼出し
- ・ 二次入口
- ・ ラベル
- ・ 分岐
- ・ 途中脱出
- ・ 節呼出し
- ・ 節出口
- ・ 復帰
- ・ プログラム実行終了

3.10.6.1 プログラム呼出し

プログラム呼出しは、別のプログラムまたは二次入口に制御を移します。

記述形式



記述規則

プログラム呼出し図記号の中には、プログラム名を記述します。

プログラム呼出し図記号の本文には、パラメタを記述します。パラメタは省略することができます。

パラメタを囲むシステム定義のキーワード「(」および「)」は、全角でも半角でも記述できます。

プログラム名

プログラム名には、COBOLのプログラム名を記述します。プログラム名の両端を引用符で囲む必要はありません。

パラメタ

パラメタは、変数の受け渡しを行う場合に指定します。

パラメタには、プログラムに渡す変数を、COBOLのUSING指定の中の作用対象の記述規則に従って記述します。ただし、語USINGは記述する必要はありません。

注意

以下に示す場合は、本表記を用いて別のプログラムまたは二次入口に制御を移すことはできません。例外処理、順次文または目的言語記述を利用して、COBOLのCALL文で記述してください。

- COBOLのON OVERFLOWを指定する
- COBOLのON EXCEPTIONを指定する
- COBOLのNOT ON EXCEPTIONを指定する
- プログラム名にCOBOLの一意名を指定する

参照

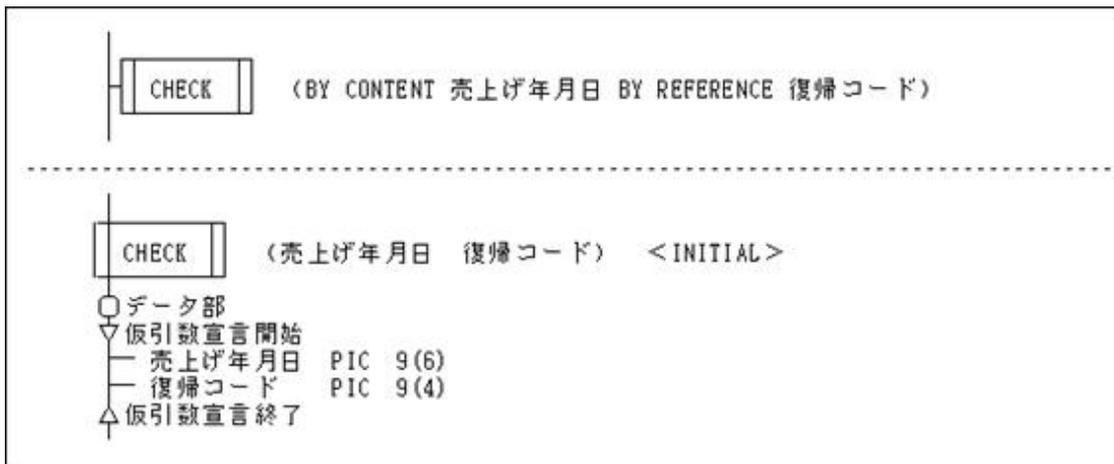
COBOLの用語

プログラム名

USING指定 ⇒ CALL文

CALL文

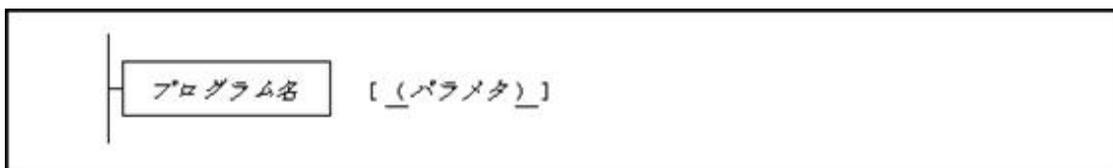
記述例



3.10.6.2 内部プログラム呼出し

内部プログラム呼出しは、内部プログラムに制御を移します。

記述形式



記述形式

内部プログラム呼出し図記号の中には、プログラム名を記述します。

内部プログラム呼出し図記号の本文には、パラメタを記述します。パラメタは省略することができます。

パラメタを囲むシステム定義のキーワード「(」および「)」は、全角でも半角でも記述できます。

プログラム名

プログラム名には、COBOLのプログラム名を記述します。プログラム名の両端を引用符で囲む必要はありません。

パラメタ

パラメタは、変数の受け渡しを行う場合に指定します。

パラメタには、内部プログラムに渡す変数を、COBOLのUSING指定の中の作用対象の記述規則に従って記述します。ただし、語USINGは記述する必要はありません。

注意

以下に示す場合は、本表記を用いて内部プログラムに制御を移すことはできません。例外処理、順次文または目的言語記述を利用して、COBOLのCALL文で記述してください。

- COBOLのON OVERFLOWを指定する
- COBOLのON EXCEPTIONを指定する
- COBOLのNOT ON EXCEPTIONを指定する
- プログラム名にCOBOLの一意名を指定する

参照

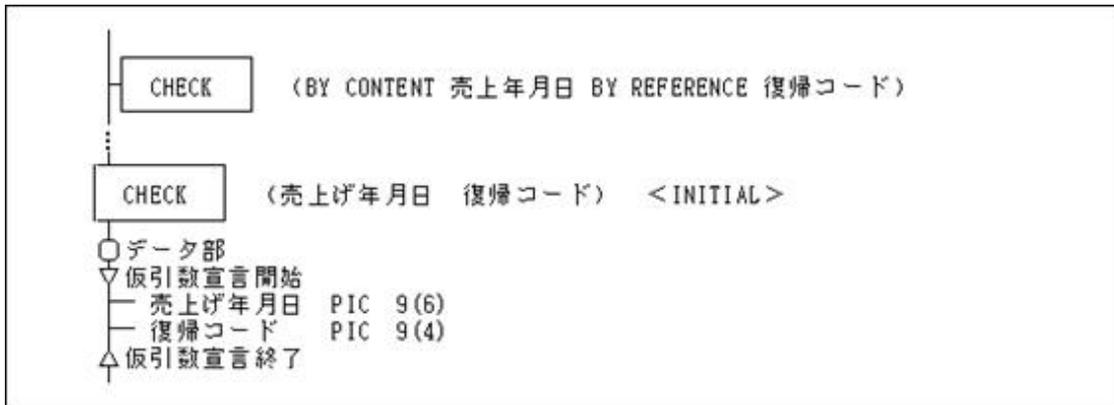
COBOLの用語

プログラム名

USING指定 ⇒ CALL文

CALL文

記述例



3.10.6.3 二次入口

二次入口は、プログラムの主入口とは別のプログラムの入口点を定めます。COBOLのENTRY文に対応する表記です。

記述形式



記述規則

二次入口は、プログラム単位の手続き部の主制御線上に記述します。内部プログラム単位の手続き部には記述できません。

二次入口図記号の中には、入口名を記述します。

二次入口図記号の本文には、パラメタを記述します。パラメタは省略することができます。

パラメタを囲むシステム定義のキーワード「(」および「)」は、全角でも半角でも記述できます。

入口名

入口名には、COBOLのプログラム名を記述します。プログラム名の両端を引用符で囲む必要はありません。

パラメタ

パラメタは、変数の受け渡しを行う場合に指定します。

パラメタは、COBOLのUSING指定の中の作用対象の記述規則に従って記述します。ただし、語USINGは記述する必要はありません。

一般規則

二次入口図記号の本文に記述したパラメタは、二次入口を含むプログラム単位の仮引数宣言節で宣言してください。



参照

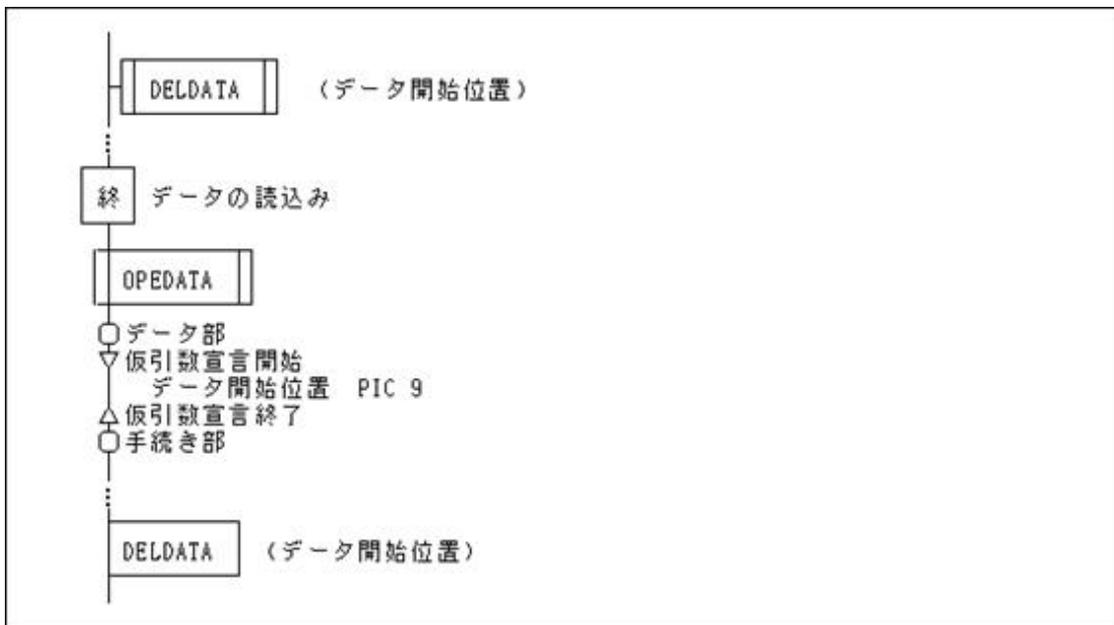
COBOLの用語

ENTRY文

プログラム名

USING指定 ⇒ CALL文

記述例



3.10.6.4 ラベル

ラベルは、手続き部の段落の始まりを示します。

記述形式



記述規則

ラベルは、手続き部の節内の主制御線上に記述します。

ラベル図記号の本文には、ラベル名を記述します。

ラベル名を囲むシステム定義のキーワード「<」および「>」は、全角でも半角でも記述できます。

ラベル名

ラベル名には、COBOLの段落名を記述します。



注意

「JVV」で始まるラベル名は、YPS/COBOLコンパイラが生成するラベル名と重複する可能性があるため、使用しないようにしてください。



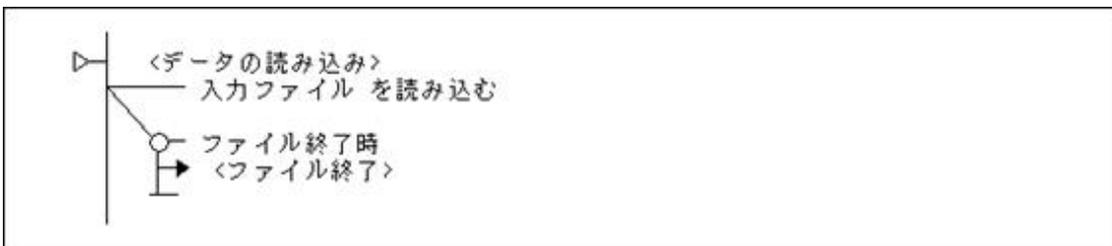
参照

COBOLの用語

段落

段落名

記述例



3.10.6.5 分岐

分岐は、指定した分岐先に無条件に分岐します。COBOLのGO TO文に対応する表記です。

記述形式



記述規則

分岐図記号の本文には、分岐先を記述します。

分岐先を囲むシステム定義のキーワード「<」および「>」は、全角でも半角でも記述できます。

分岐先

分岐先には、COBOLの手続き名を記述します。

注意

COBOLのGO TO文でDEPENDING ON指定を行いたい場合、または分岐先を省略したい場合には、本表記を利用しないで、順次文または目的言語記述を利用して、COBOLのGO TO文を記述してください。

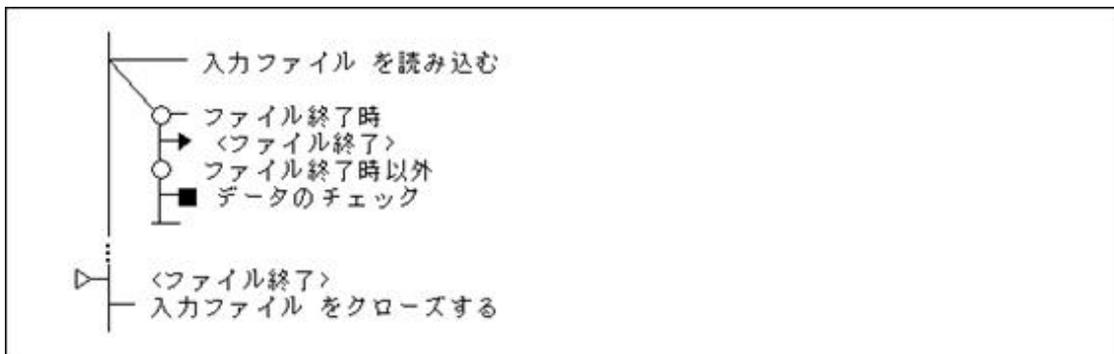
参照

COBOLの用語

GO TO文

手続き名

記述例



3.10.6.6 途中脱出

途中脱出は、入れ子になったループ系の表記および選択系の表記の処理の途中で、主制御線上の処理に制御を移します。

記述形式

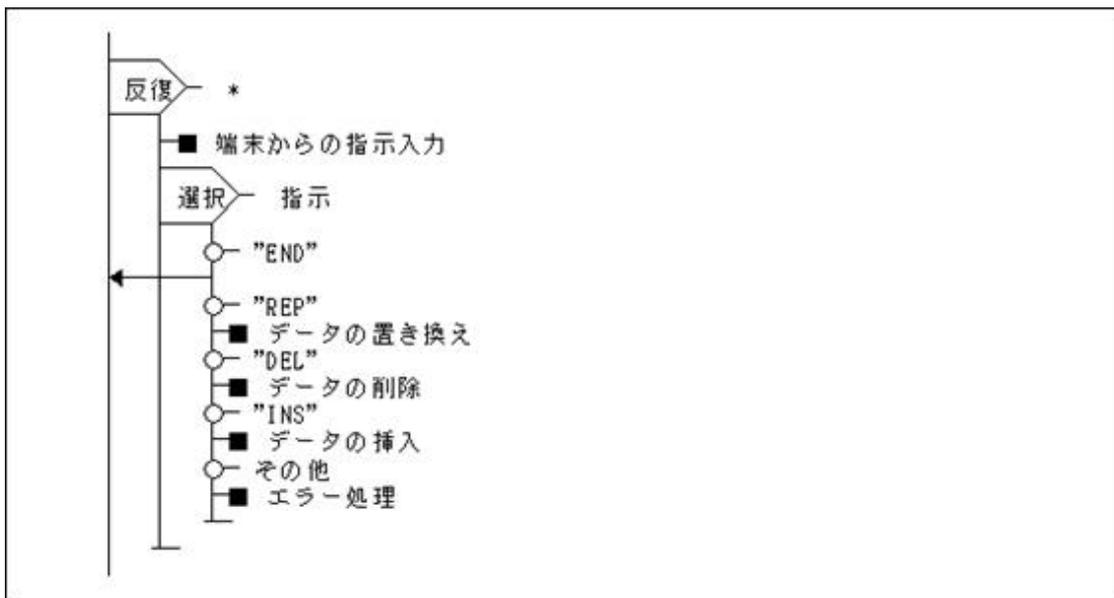


記述規則

途中脱出は、ループ系の表記および選択系の表記内に記述します。ループ系の表記および選択系の表記に含まれる表記内に記述することもできます。

途中脱出の脱出先(矢印が指し示す制御線)には、主制御線を指定します。

記述例



3.10.6.7 節呼出し

節呼出しは、指定した節の処理を実行します。COBOLの1つの節名を指定したそとPERFORM文に対応する表記です。

記述形式



記述規則

節呼出し図記号の本文には、節名を記述します。

節名

節名には、COBOLの節名を記述します。

注意

以下に示すそとPERFORM文は、本表記を用いて表現することはできません。順次文または目的言語記述を利用して、COBOLのPERFORM文で記述してください。

- THRU指定を行う
- 手続き名に段落名を指定する

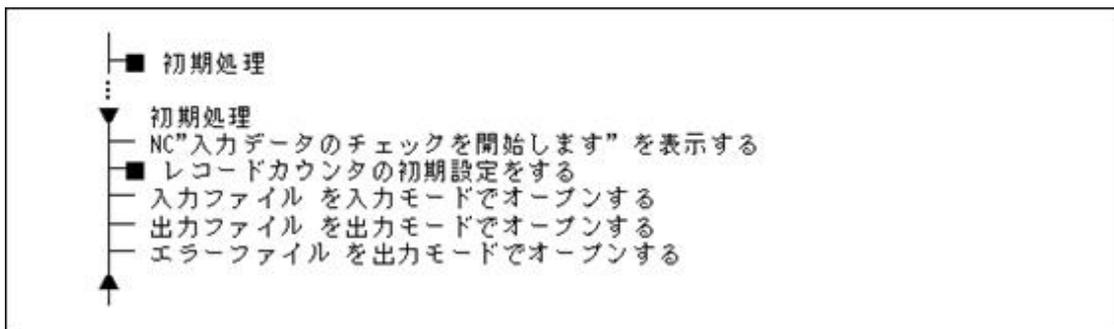
参照

COBOLの用語

そとPERFORM文 ⇒ PERFORM文

節名

記述例



3.10.6.8 節出口

節出口は、節出口が含まれる節の最後に制御を移行します。

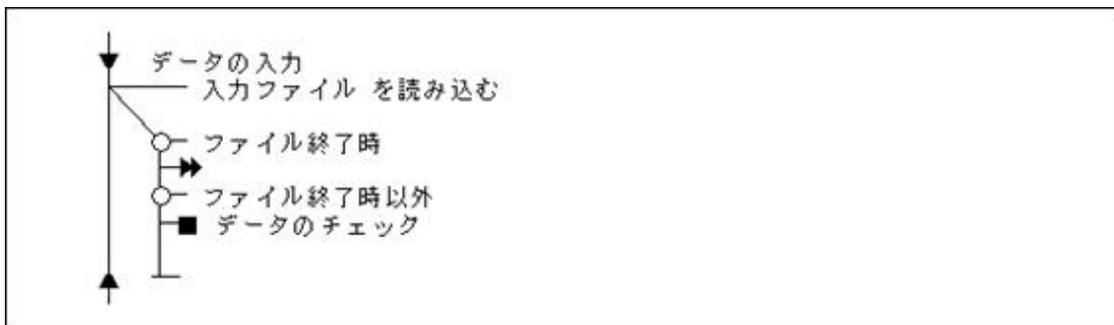
記述形式



記述規則

節出口は、手続き部の節内に記述します。ただし、節開始および節終了を省略した節内には、記述することができません。

記述例



3.10.6.9 復帰

復帰は、呼出し元のプログラムに制御を戻します。

記述形式



記述規則

復帰図記号の本文には、復帰値を記述します。復帰値は省略することができます。

復帰値

復帰値には、COBOLの算術式を記述します。

一般規則

復帰値(COBOLの算術式の評価結果)が、特殊レジスタPROGRAM-STATUSに設定されます。復帰値を省略した場合には、特殊レジスタPROGRAM-STATUSには何も設定されません。



参照

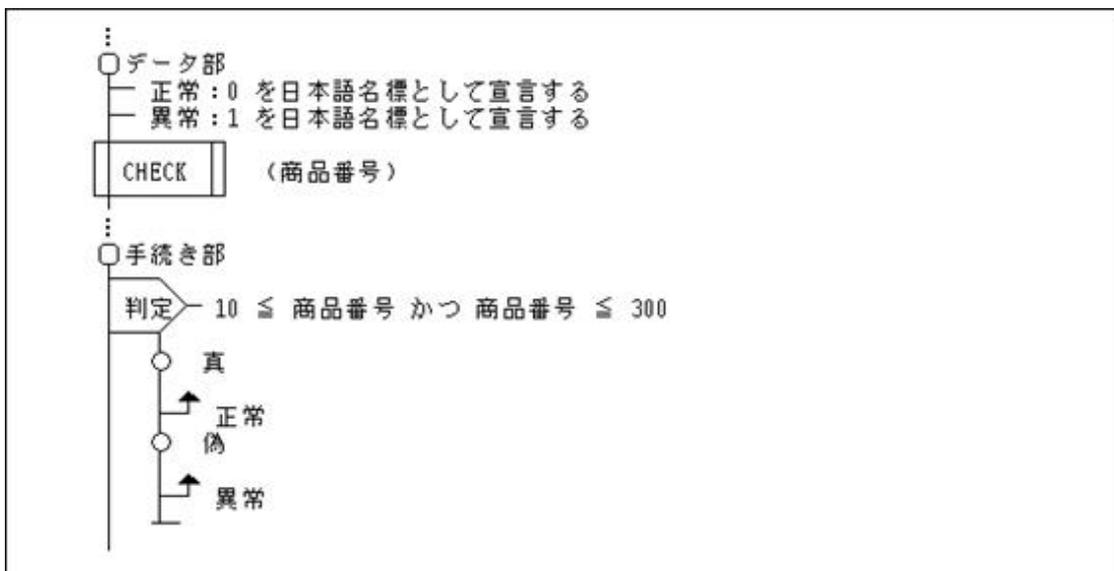
COBOLの用語

算術式

特殊レジスタ

PROGRAM-STATUS

記述例



3.10.6.10 プログラム実行終了

プログラム実行終了は、プログラムの実行を終了します。

記述形式



記述規則

実行終了図記号の本文には、復帰値を記述します。復帰値は省略することができます。

復帰値

復帰値には、COBOLの算術式を記述します。

一般規則

復帰値(COBOLの算術式の評価結果)が、特殊レジスタPROGRAM-STATUSに設定されます。復帰値を省略した場合には、特殊レジスタPROGRAM-STATUSには何も設定されません。

COBOLの用語

算術式

特殊レジスタ

PROGRAM-STATUS

記述例



3.11 内部プログラム単位を構成する表記

内部プログラム単位を構成する表記は、以下の点を除いてプログラム単位と同じです。プログラム単位を構成する表記を参照してください。

- ・ 内部プログラム単位的环境部には、システム環境宣言節を記述することはできません。
- ・ 内部プログラム単位の手続き部の節には、二次入口を記述することはできません。

3.12 部および節に依存しない表記

部および節に依存しない表記には、以下の5種類があります。

記述できる場所については、各表記の記述規則を参照してください。

- ・ 注釈
- ・ 要約定義
- ・ コメント文
- ・ 目的言語記述
- ・ 強制改ページ

3.12.1 注釈

注釈は、表記を翻訳の対象から無条件に除外します。

記述形式



記述規則

注釈は、目的言語記述内を除く、モジュール本体内の任意の場所に記述することができます。

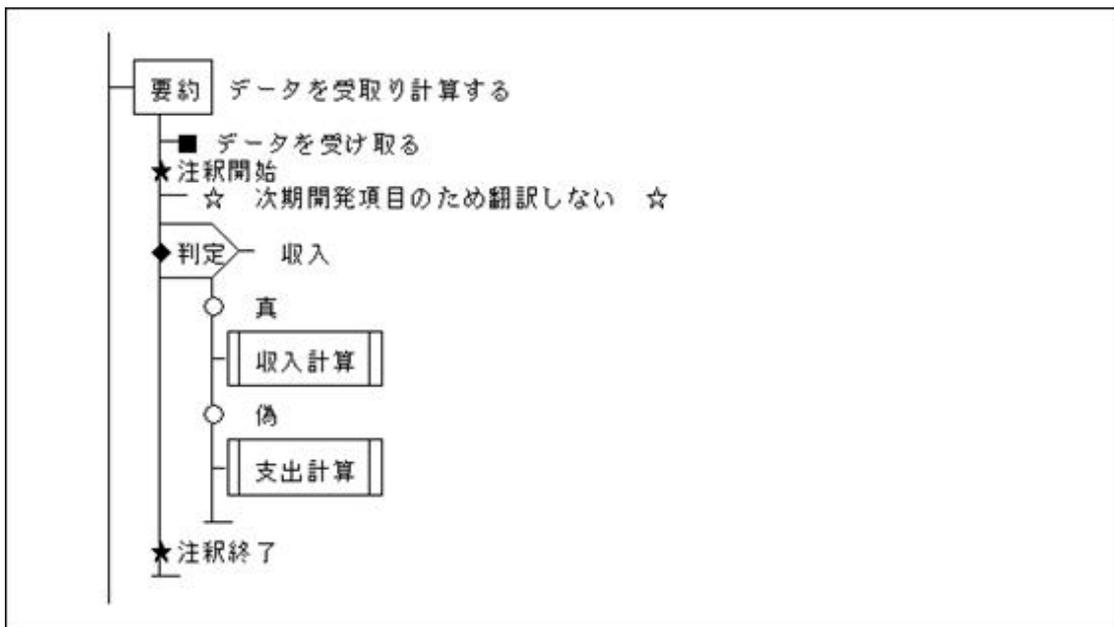
注釈開始と注釈終了は、途切れていない物理的な同一制御線上に、対応させて記述します。

注釈内には、注釈開始と注釈終了を除く任意の表記を記述することができます。

一般規則

注釈が存在した場合、注釈開始の前と注釈終了の後ろを結合したYPS仕様書が翻訳の対象となります。よって、注釈内に不完全な表記が存在しても、翻訳には差し支えありません。

記述例



3.12.2 要約定義

要約定義は、一連の処理や宣言に対して、その概要を説明するためのコメントを記述します。

記述形式



記述規則

要約定義は、構文定義部、モジュール共通データ部、環境部、プログラム内データ部および手続き部に記述することができます。

要約図記号の本文には、要約文を記述します。要約文は省略することができます。

要約定義内には以下の表記を記述することができます。

- ・ 条件付き翻訳判定、条件付き翻訳単一選択
- ・ 取込み

- ・ コメントキーワード定義文
- ・ 抽象化処理構文定義、一時変数宣言文、抽象化条件式定義文、抽象化処理構文の引用文
- ・ 日本語名標宣言文
- ・ 環境部を構成する表記、環境部の節を構成する表記
- ・ プログラム内データ部を構成する表記、データ部の節を構成する表記
- ・ 手続き部を構成する表記、手続き部の節を構成する表記
- ・ 部および節に依存しない表記

ただし、以下の表記を記述する場合には、開始と終了を対応させて記述してください。

- ・ 抽象化処理構文定義
- ・ 環境部、プログラム内データ部および手続き部の節
- ・ 宣言部分

要約文

要約文は、一連の処理や宣言に対して、その概要を説明するために記述するコメントです。

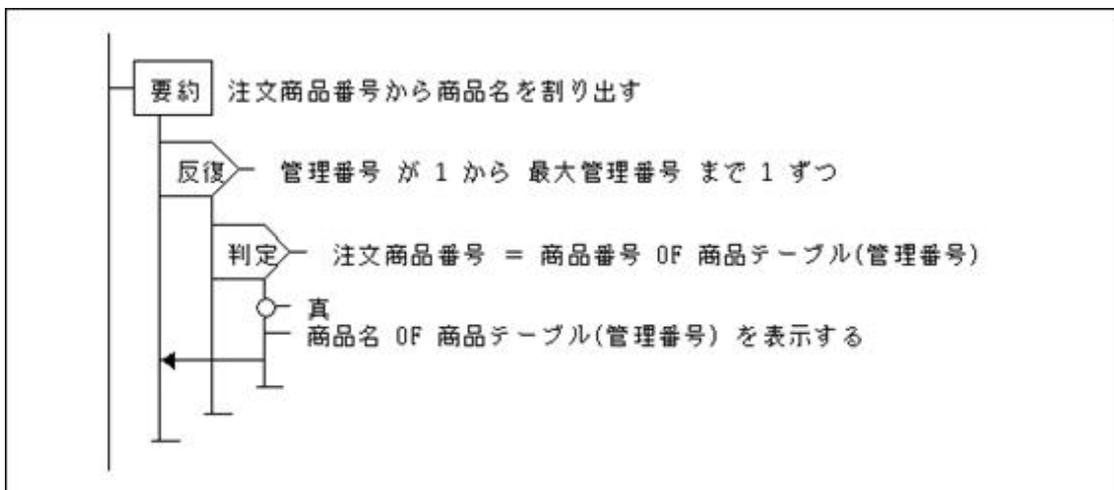
要約文は、YPS仕様書で使用できる文字で記述します。



注意

要約定義を物理的な主制御線上に記述した場合、要約定義から伸びる制御線が論理的な主制御線になります。したがって、要約定義内に途中脱出を記述する場合には、脱出先には要約定義から伸びる制御線を指定してください。

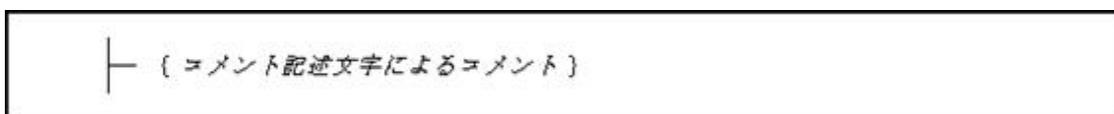
記述例



3.12.3 コメント文

コメント文は、コメントを記述する文です。

記述形式

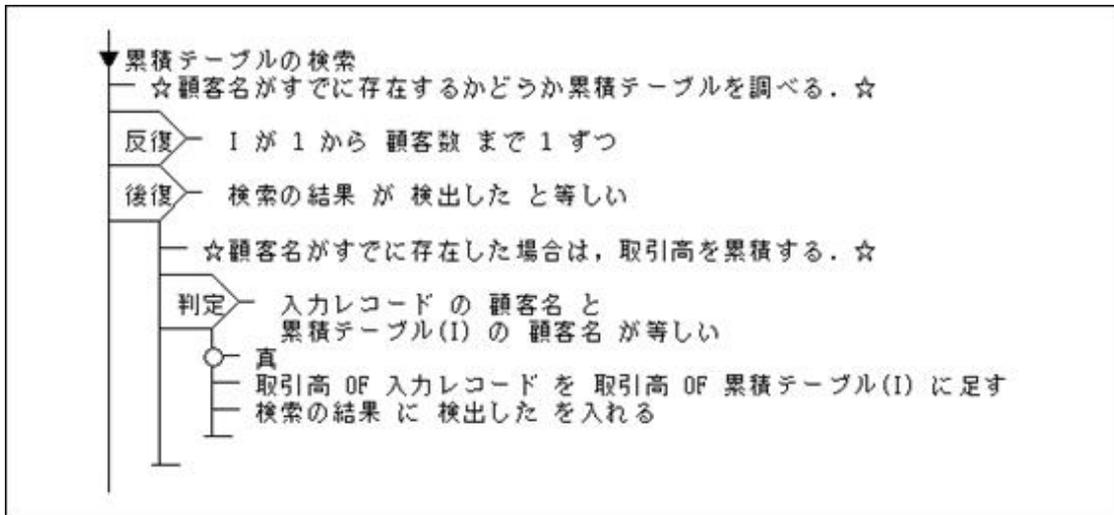


記述規則

コメント文は、目的言語記述内を除く、モジュール本体内の任意の場所に記述することができます。

コメント文の本文には、コメント記述文字によるコメントを記述します。コメント記述文字によるコメントについては、“[2.6.1 YPSの文字列](#)”を参照してください。

記述例

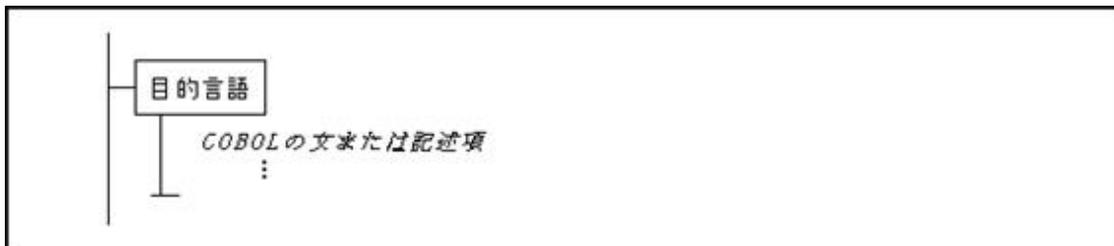


3.12.4 目的言語記述

目的言語記述は、YPS仕様書内にCOBOL原始プログラムを記述します。

本表記は、YPS/COBOL言語の表記では記述できないCOBOL原始プログラムを記述する場合に使用します。

記述形式



記述規則

目的言語記述は、抽象化処理構文定義の定義体、環境部の節、データ部の節および手続き部の節内に記述することができます。

目的言語記述の継続行に、COBOLの文または記述項を記述します。継続行の一行がCOBOLの一行に、継続行の先頭がCOBOLの7カラム目(標識領域)に対応します。

目的言語記述内にコメントを記述する場合には、COBOLの文法に従ってコメントを記述します。

目的言語記述内は、すべてのYPSの機能の作用対象外です。したがって、日本語名標宣言文で宣言した文字列1を引用したり、全角文字記述機能を使用して、COBOLでは半角文字で記述しなければならない文字を全角文字で記述したりすることはできません。

記述例



3.12.5 強制改ページ

強制改ページは、印刷時に強制的に改ページを行うことを指定します。

記述形式

[形式1(モジュール概要記述の場合)]



[形式2(モジュール本体の場合)]



記述規則

強制改ページは、モジュール概要記述の任意の場所、および目的言語記述を除くモジュール本体の任意の場所に記述することができます。ただし、モジュール本体に記述する場合、継続行に記述することはできません。

注意

改ページ図記号「◎」は必ずYPS表記編集で指定してください。全角文字の「◎」は強制改ページ表記とはみなされません。

YPS仕様書を印刷する製品(YPSドキュメントなど)により、オプションで強制改ページを有効にするかどうかを選択できる製品、無条件に強制改ページを無視する製品があります。強制改ページが有効になるかどうかは、印刷時に使用する製品の使用手引書で確認してください。

記述例

[形式1]

TRANS, 取引高の累積, V12L10, 富士通, 2018-03-01
◆ COPYRIGHT 1990-2018 FUJITSU LIMITED
◇ 翻訳: 引用符(複), 全角文字記述(あり)
◎
【機能】
入力ファイルに入っているレコード(顧客名と取引高の組)を全件読み込み,
同じ顧客名の取引高を累積して一つのレコードにして, 出力ファイルに書き出す。
ただし, 当モジュールの仕様は, 以下のとおりとする。
- 顧客数は, 500社以下である。
- 総取引高は, 8桁以下の値である。

[形式2]

■ 前処理
■ 整列処理
■ 索引ファイル創成処理
■ 終了処理
◎
▼ 前処理

第4章 条件付き翻訳機能

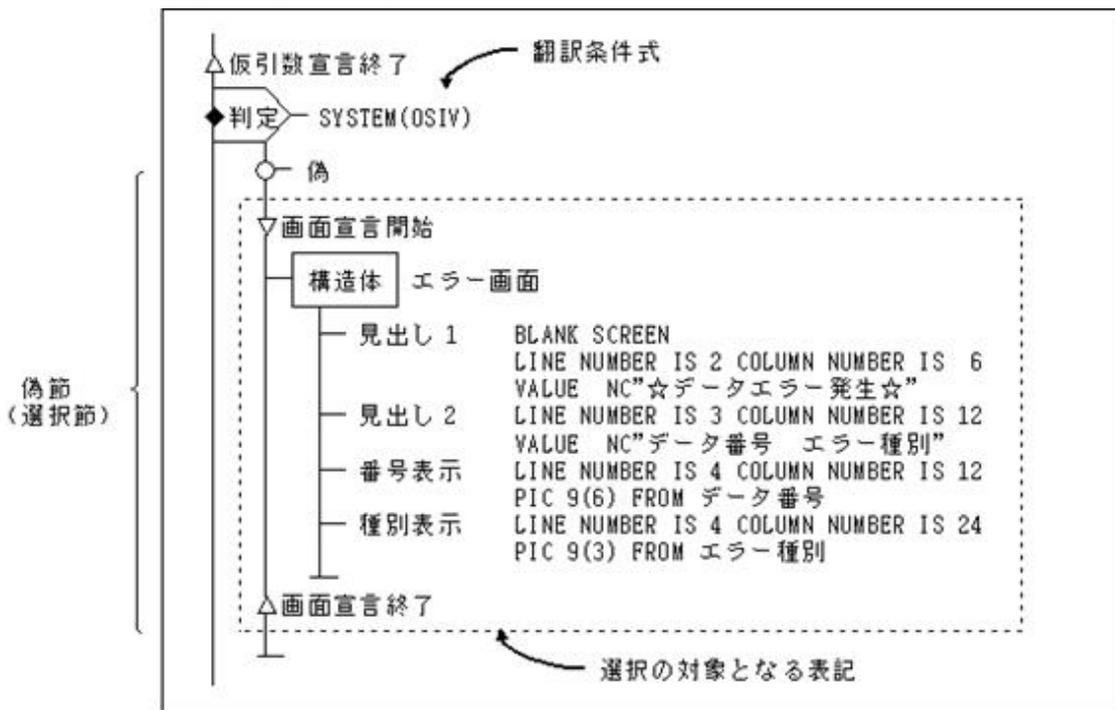
本章では、条件付き翻訳機能について説明します。

4.1 条件付き翻訳機能とは

条件付き翻訳機能とは、翻訳時のオプション(指定文字列)により、翻訳の対象とする表記を選択する機能です。条件付き翻訳機能を用いることにより、対象システムの違いなどにより、処理および宣言を変更する必要があるプログラムの、仕様書の一元管理が行えます。

条件付き翻訳機能の使用例を“[図4.1 条件付き翻訳機能の使用例](#)”に示します。この例では、画面宣言節がGS/Mシリーズではサポートされていないため、対象のシステムがGS/M以外の時だけ翻訳の対象にしようとしています。

図4.1 条件付き翻訳機能の使用例



表記の種類

条件付き翻訳には、以下に示す表記があります。目的に応じて使い分けてください。

条件付き翻訳判定

翻訳条件式を評価し、その評価結果により翻訳の対象とする表記を選択します。“[図4.1 条件付き翻訳機能の使用例](#)”の使用例では、この表記を使用しています。詳細については“[4.3 条件付き翻訳判定](#)”を参照してください。

条件付き翻訳単一選択(選択条件名指定)

翻訳時のオプションで指定された選択条件名と条件指定節の本文に記述された選択条件名を上から比較し、最初に文字列が一致した条件指定節の表記を翻訳の対象にします。詳細については“[4.4 条件付き翻訳単一選択\(選択条件名指定\)](#)”を参照してください。

条件付き翻訳単一選択(翻訳条件式指定)

翻訳条件式を上から評価し、最初に翻訳条件式が真になった条件指定節の表記を翻訳の対象にします。詳細については“[4.5 条件付き翻訳単一選択\(翻訳条件式指定\)](#)”を参照してください。

記述できる場所

条件付き翻訳は、モジュール本体の任意の場所に記述することができます。ただし、目的言語記述内には記述することはできません。

選択節に記述できる表記

条件付き翻訳の選択節には、選択節を除く任意の表記を記述することができます。

ただし、以下の表記を記述する場合には、開始と終了を対応させて記述してください。

- ・ 抽象化処理構文定義
- ・ 環境部、プログラム内データ部および手続き部の節
- ・ 宣言部分

翻訳オプションの指定方法

翻訳オプションでの翻訳条件名および選択条件名の指定方法は、システムにより異なります。各システムの使用手引書を参照してください。

4.2 翻訳条件式

翻訳条件式は、翻訳の対象にする条件を規定します。

記述形式

[基本形]

翻訳条件名 [(選択条件名)]

[応用形1(論理演算子による結合)]

翻訳条件式 (かつ または) 翻訳条件式

[応用形2(結合順序の変更)]

(翻訳条件式)

記述規則

翻訳条件式には、翻訳条件名と選択条件名の組合せを記述します。選択条件名は、システム定義のキーワード「(」および「)」で囲みます。選択条件名は、省略することができます。

選択条件名を囲むシステム定義のキーワード「(」および「)」は、全角でも半角でも記述できます。

論理演算子による結合

翻訳条件式をシステム定義のキーワード「かつ」または「または」を用いて結合することができます。

結合順序の変更

翻訳条件式をシステム定義のキーワード「(」および「)」で囲むことにより、結合順序を変更することができます。

結合順序を変更するシステム定義のキーワード「(」および「)」は、全角でも半角でも記述できます。

翻訳条件名

翻訳条件名には、1文字以上30文字以内の全角または半角の文字列を記述します。文字列には、全角および半角の空白、「(」,「)」および「,」を含めることはできません。

選択条件名

選択条件名には、1文字以上30文字以内の全角または半角の文字列を記述します。文字列には、全角および半角の空白、「(」、「)」および「,」を含めることはできません。

記述例

[基本形]

```
DEBUG
SYSTEM(UXP/DS)
```

[応用形1(論理演算子による結合)]

```
SYSTEM(UXP/DS) または SYSTEM(WINDOWS)
```

[応用形2(結合順序の変更)]

```
( SYSTEM(UXP/DS) または SYSTEM(WINDOWS) )かつ DEBUG
```

翻訳条件式の評価方法

翻訳条件式は、以下に示した条件を満たしている場合に真になります。

- YPS仕様書に記述した翻訳条件式が翻訳条件名だけの場合
 - 翻訳時のオプションに翻訳条件名だけが指定されている。かつ、
 - 翻訳時のオプションに指定した翻訳条件名がYPS仕様書に記述した翻訳条件名と同一の文字列である場合
- YPS仕様書に記述した翻訳条件式が翻訳条件名と選択条件名の場合
 - 翻訳時のオプションに翻訳条件名および選択条件名が指定されている。かつ、
 - 翻訳時のオプションに指定した翻訳条件名がYPS仕様書に記述した翻訳条件名と同一の文字列である。かつ、
 - 翻訳時のオプションに指定した選択条件名がYPS仕様書に記述した選択条件名と同一の文字列である場合

論理演算子による結合

システム定義のキーワード「かつ」は論理積を、「または」は論理和を表します。

論理積を表す「かつ」の方が、論理和を表す「または」より先に結合されます。

結合順序の変更

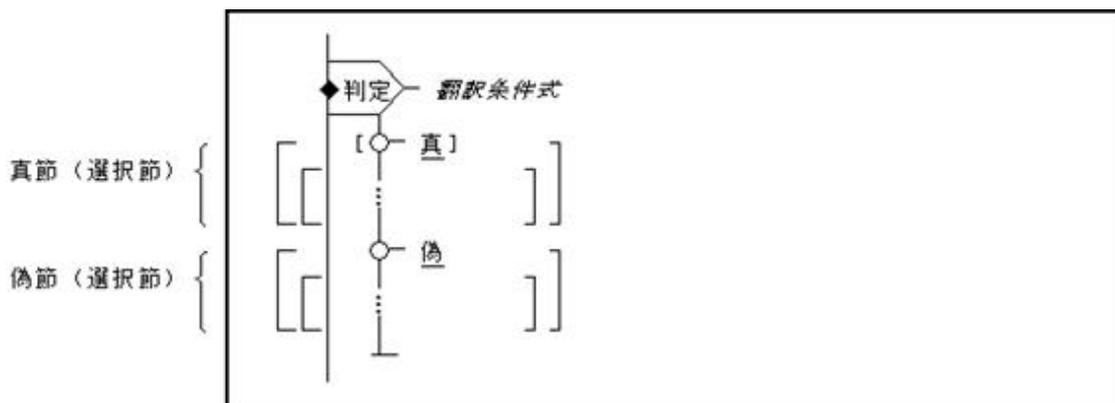
結合順序を変更するシステム定義のキーワード「(」と「)」で囲んだ翻訳条件式は、「(」と「)」で囲んでいない翻訳条件式より先に結合されます。

4.3 条件付き翻訳判定

条件付き翻訳判定は、翻訳条件式を評価し、その評価結果により翻訳の対象とする表記を選択します。

翻訳条件式が真の場合には、真節があれば真節の表記を翻訳の対象にし、なければ何も翻訳の対象にしません。翻訳条件式が偽の場合には、偽節があれば偽節の表記を翻訳の対象にし、なければ何も翻訳の対象にしません。

記述形式



記述規則

条件付き翻訳判定図記号の本文には、翻訳条件式を記述します。記述規則については“[4.2 翻訳条件式](#)”を参照してください。

条件付き翻訳判定には、真節と偽節を記述形式に示した順序で記述します。真節または偽節のどちらか一方を省略することができます。

真節

真節は、翻訳条件式の評価結果が真の場合に翻訳の対象とする表記を記述するところです。

真節の開始を表す丸図記号の本文には、システム定義のキーワード「真」、「YES」または「yes」を記述します。「YES」および「yes」は全角でも半角でも記述できます。

偽節を省略する場合には、真節の開始を表す「○ 真」を省略することができます。

真節に記述できる表記については“[4.1 条件付き翻訳機能とは](#)”を参照してください。

偽節

偽節は、翻訳条件式の評価結果が偽の場合に翻訳の対象とする表記を記述するところです。

偽節の開始を表す丸図記号の本文には、システム定義のキーワード「偽」、「NO」または「no」を記述します。「NO」および「no」は全角でも半角でも記述できます。

偽節に記述できる表記については“[4.1 条件付き翻訳機能とは](#)”を参照してください。

一般規則

翻訳条件式の評価方法については“[4.2 翻訳条件式](#)”を参照してください。

記述例

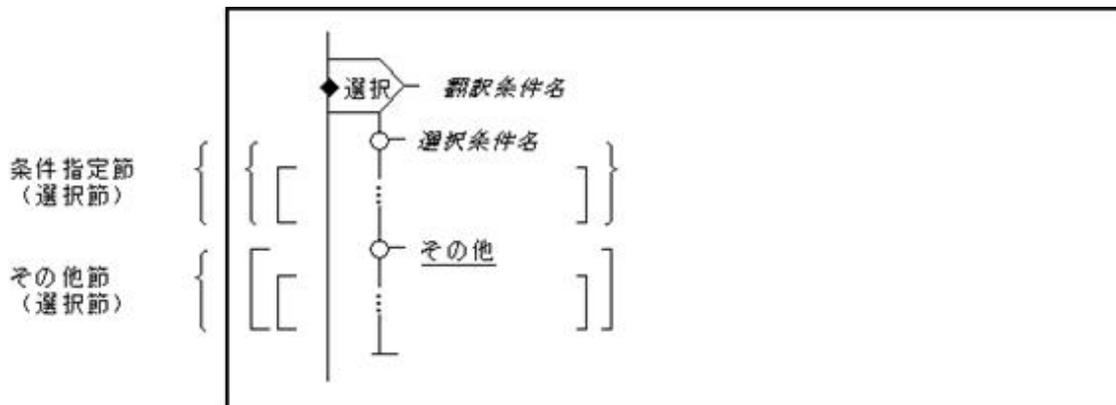


備考: プログラム名を日本語で記述する場合、日本語名標宣言文であらかじめ宣言しておく必要があります。

4.4 条件付き翻訳単一選択(選択条件名指定)

条件付き翻訳単一選択(選択条件名指定)は、翻訳時のオプションで指定された選択条件名と条件指定節の本文に記述された選択条件名を上から比較し、最初に文字列が一致した条件指定節の表記を翻訳の対象にします。文字列が一致する条件指定節がなかった場合には、その他節があればその他節の表記を翻訳の対象にし、なければ何も翻訳の対象にしません。

記述形式



記述規則

条件付き翻訳選択図記号の本文には、翻訳条件名を記述します。記述規則については“4.2 翻訳条件式”を参照してください。

条件付き翻訳選択(選択条件名指定)には、1つ以上の条件指定節とその他節を記述形式で示した順序で記述します。その他節は省略することができます。

条件指定節

条件指定節は、翻訳時のオプションで指定された選択条件名と条件指定節の本文に記述された選択条件名の文字列が一致した場合に翻訳の対象とする表記を記述するところです。

条件指定節の開始を表す丸図記号の本文には、選択条件名を記述します。記述規則については“4.2 翻訳条件式”を参照してください。

条件指定節に記述できる表記については“4.1 条件付き翻訳機能とは”を参照してください。

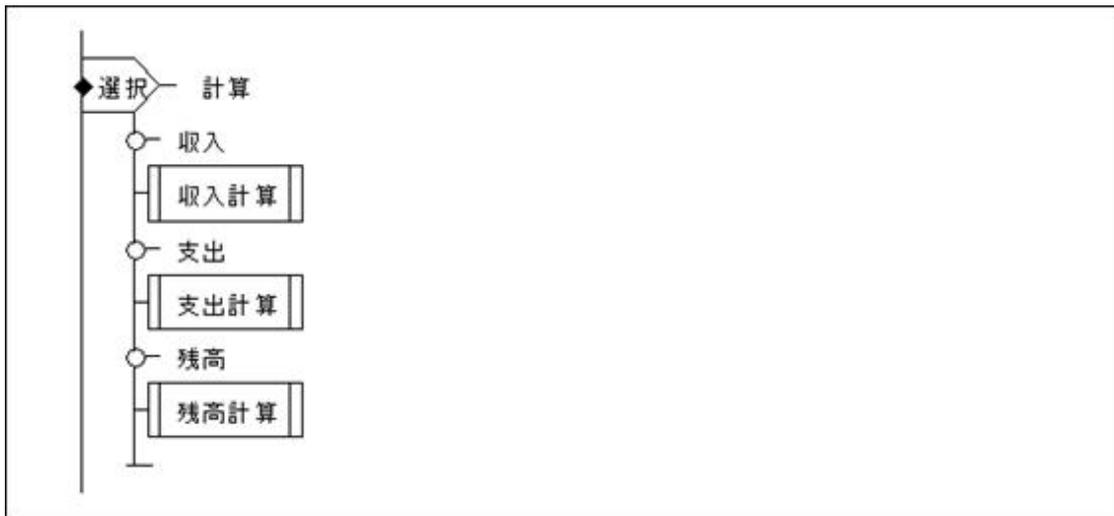
その他節

その他節は、文字列が一致する条件指定節がなかった場合に翻訳の対象とする表記を記述するところです。

その他節の開始を表す丸図記号の本文には、システム定義のキーワード「その他」、「OTHER」または「other」を記述します。「OTHER」および「other」は全角でも半角でも記述できます。

その他節に記述できる表記については“4.1 条件付き翻訳機能とは”を参照してください。

記述例

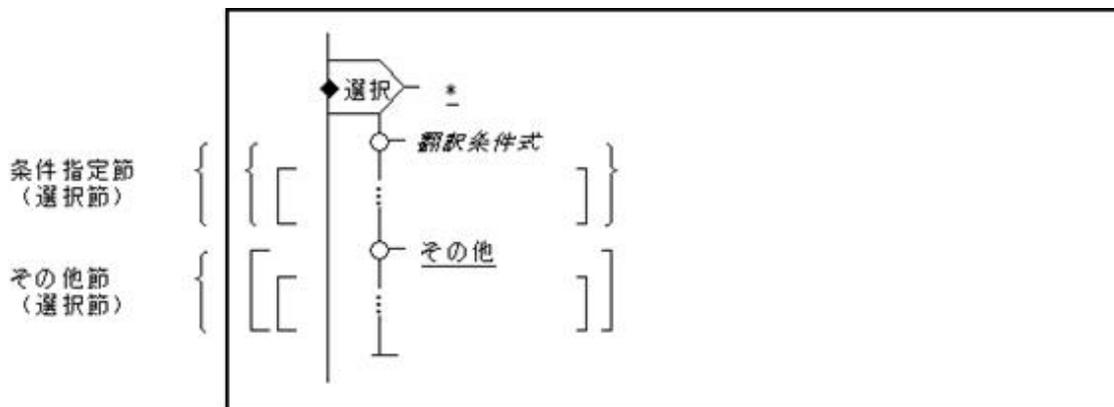


備考: プログラム名を日本語で記述する場合、日本語名標宣言文であらかじめ宣言しておく必要があります。

4.5 条件付き翻訳単一選択(翻訳条件式指定)

条件付き翻訳単一選択(翻訳条件式指定)は、翻訳条件式を上から評価し、最初に翻訳条件式が真になった条件指定節の表記を翻訳の対象にします。翻訳条件式が真である条件指定節がなかった場合には、その他節があればその他節の表記を翻訳の対象にし、なければ何も翻訳の対象にしません。

記述形式



記述規則

条件付き翻訳選択図記号の本文には、システム定義のキーワード「*」を記述します。「*」は全角でも半角でも記述できます。

条件付き翻訳単一選択(翻訳条件式指定)には、1つ以上の条件指定節とその他節を記述形式に示した順序で記述します。その他節は省略することができます。

条件指定節

条件指定節は、翻訳条件式が真になった場合に翻訳の対象とする表記を記述するところです。

条件指定節の開始を表す丸図記号の本文には、翻訳条件式を記述します。記述規則については“[4.2 翻訳条件式](#)”を参照してください。

条件指定節に記述できる表記については“[4.1 条件付き翻訳機能とは](#)”を参照してください。

その他節

その他節は、翻訳条件式が真になる条件指定節がなかった場合に翻訳の対象とする表記を記述するところです。

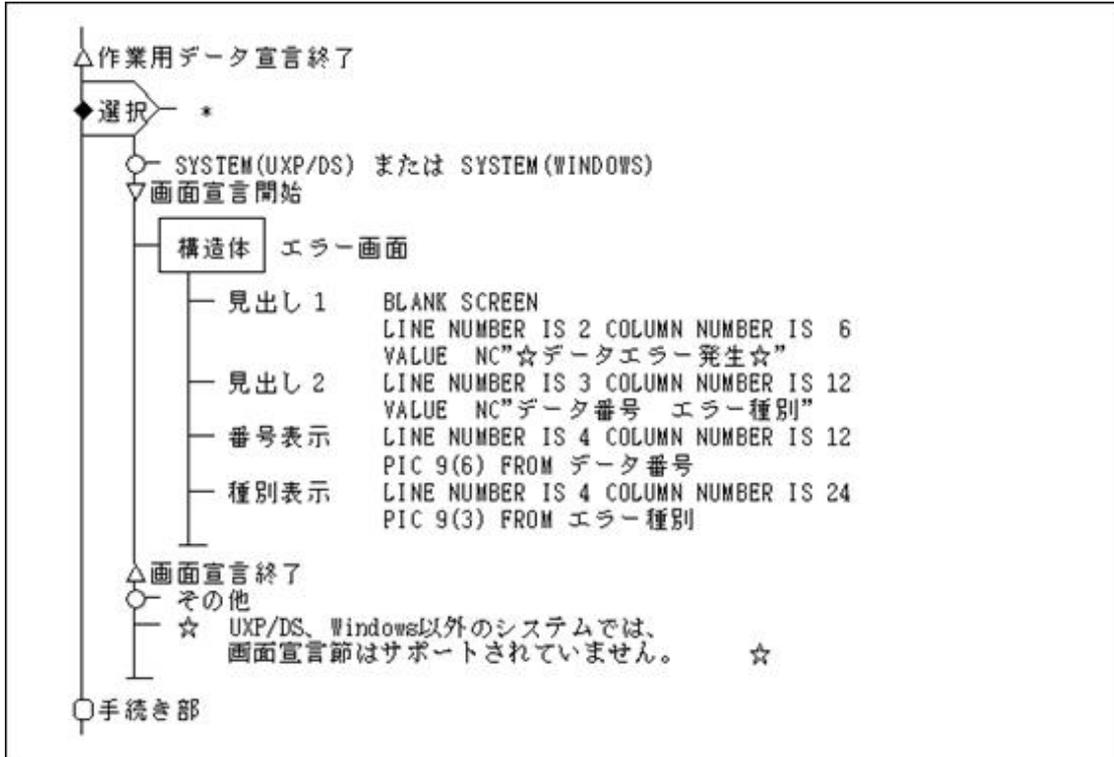
その他節の開始を表す丸図記号の本文には、システム定義のキーワード「その他」、「OTHER」または「other」を記述します。「OTHER」または「other」は全角でも半角でも記述できます。

その他節に記述できる表記については“4.1 条件付き翻訳機能とは”を参照してください。

一般規則

翻訳条件式の評価方法については“4.2 翻訳条件式”を参照してください。

記述例



第5章 取込み機能

本章では、取込み機能について説明します。

5.1 取込み機能とは

取込み機能とは、プログラムの一部を他の仕様書から取り込む機能です。取込み機能を使用することにより、処理および宣言の部品化が行えます。

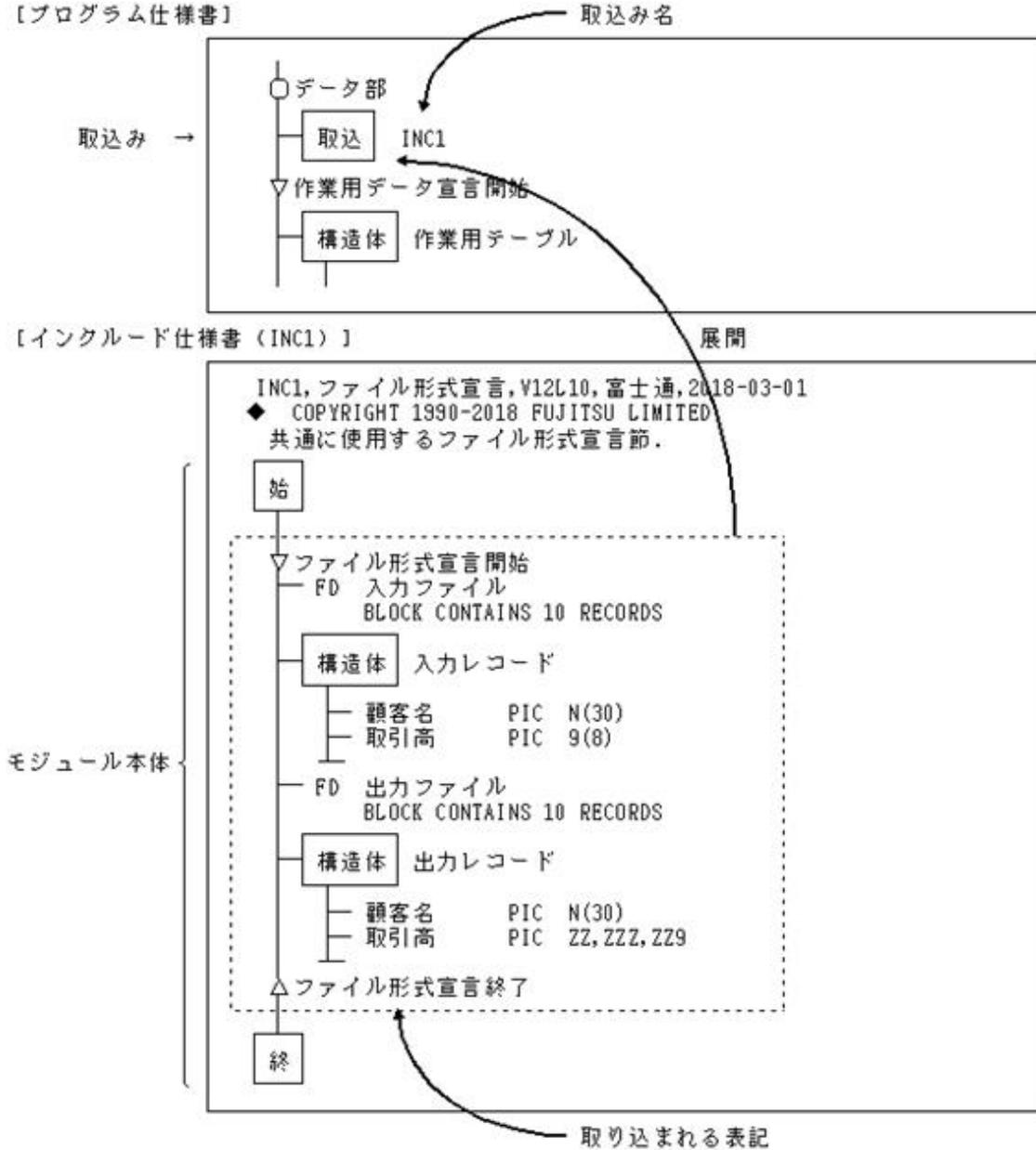
取込み機能を実現するためには、インクルード仕様書と取込み表記を使用します。

インクルード仕様書は、翻訳時に取り込まれる仕様書です。インクルード仕様書のモジュール本体に記述した表記が、取込み表記を記述した場所に展開されます。

取込みは、インクルード仕様書を取り込む場所と、取り込むインクルード仕様書の名前を表す表記です。置換語句宣言を指定することにより、インクルード仕様書内の文字列および文字列の一部を他の文字列に置き換えることができます。

“[図5.1 取込み機能の使用例](#)”に取込み機能の使用例を示します。この例では、各プログラムで共通に使用するファイル形式宣言節を取り込んでいます。

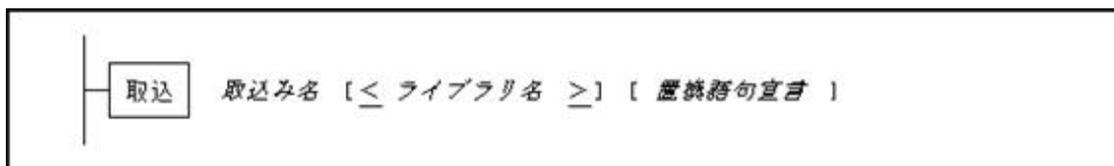
図5.1 取込み機能の使用例
 【プログラム仕様書】



5.2 取込み

取込みは、取込み名で指定されたインクルード仕様書のモジュール本体内の表記を取り込みます。取込み時にインクルード仕様書内の文字列および文字列の一部を、他の文字列に置き換えることもできます。

記述形式



記述規則

取込みは、プログラム仕様書の構文定義部、モジュール共通データ部、環境部、プログラム内データ部および手続き部に記述することができます。また、インクルード仕様書、外部構文定義仕様書および日本語名標宣言仕様書のモジュール本体にも記述することができます。ただし、文字列置換えを行う(置換語句宣言を指定した)取込みによって取り込まれるインクルード仕様書内には記述することはできません。

取込み図記号の本文には、取込み名、ライブラリ名および置換語句宣言を記述形式で示した順序で記述します。ライブラリ名と置換語句宣言は省略することができます。置換語句宣言については“5.3 置換語句宣言”を参照してください。

ライブラリ名を囲むシステム定義のキーワード「<」および「>」は、全角でも半角でも記述できます。

取込み名

取込み名は、取り込むインクルード仕様書のファイル名を表します。

ファイル名の記述規則は、システムにより異なります。各システムの使用手引書を参照してください。

ライブラリ名

ライブラリ名は、インクルード仕様書が格納されているライブラリを表します。

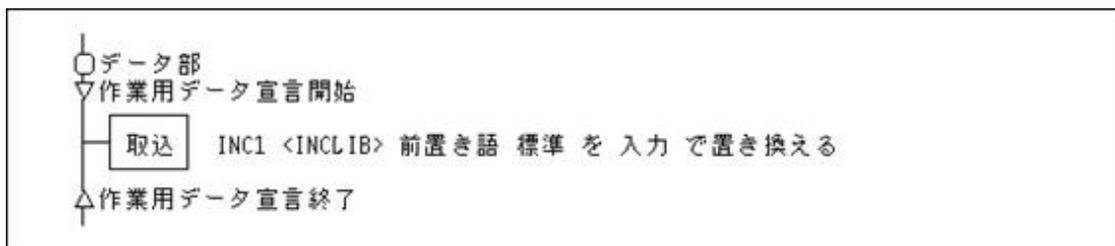
ライブラリ名には、英字で始まる8文字以内の英数字を記述します。ライブラリが各システムの何に対応するかについては、各システムの使用手引書を参照してください。

一般規則

ライブラリ名を省略した場合には、標準のライブラリが指定されたものとみなされます。

標準のライブラリについては、各システムの使用手引書を参照してください。

記述例



5.3 置換語句宣言

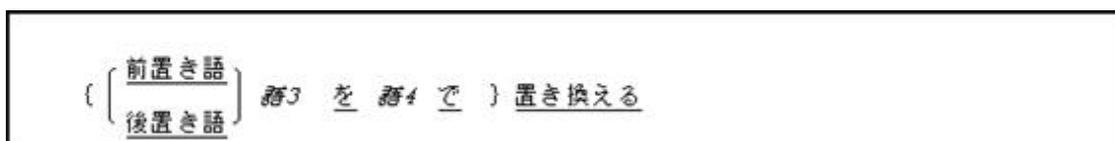
置換語句宣言は、インクルード仕様書の取込み時に置き換える文字列を宣言します。

記述形式

[形式1]



[形式2]



記述規則

置換語句宣言は、取込み表記の本文に1つだけ記述することができます。取込み表記内での記述場所については“5.2 取込み”を参照してください。

置き換える文字列は、複数指定することができます。複数指定する場合には、語1と語2の組または語3と語4の組を、繰り返して記述します。

システム定義のキーワード「を」、「で」、「前置き語」および「後置き語」の前後には、全角または半角の空白を1つ以上記述してください。ただし、「で」と「置き換える」の間の空白は省略することができます。

語1

語1には、1文字以上30文字以下の文字列を指定します。文字列は、以下のいずれかでなければなりません。

- 半角の英数字だけからなる文字列
- 空白および特殊文字を含まない全角の文字列
- 半角の「(」と「)」だけからなる文字列
- 全角の「(」と「)」だけからなる文字列

語2、語3および語4

語2、語3および語4には、1文字以上30文字以下の文字列を指定します。文字列は、以下のいずれかでなければなりません。

- 半角の英数字だけからなる文字列
- 空白および特殊文字を含まない全角の文字列

特殊文字

ここでいう特殊文字とは、以下の文字を指しています。

+(正符号)	*(アスタリスク)	/ (斜線)	= (等号)	¥ (円記号)	, (カンマ)
;(セミコロン)	. (ピリオド)	" (二重引用符)	((始め丸括弧)) (終わり丸括弧)	
> (不等号より大)	< (不等号より小)	: (コロン)	% (パーセント)	_ (下線)	? (疑問符)
' (引用符)	@ (単位記号)	# (井げた)	\$ (ドル記号)		

一般規則

[形式1]の場合、取り込まれたインクルード仕様書の本文中の、語1と同じ文字列が語2に置き換えられます。

[形式2]の場合、取り込まれたインクルード仕様書の本文中の、全角または半角の「-」を含む文字列の一部が置換えの対象になります。システム定義のキーワード「前置き語」を指定した場合には、語3と同じ「-」の左端の文字列が語4に置き換えられます。システム定義のキーワード「後置き語」を指定した場合には、語3と同じ「-」の右端の文字列が語4に置き換えられます。

置換え規則の詳細については、「置換えの手順」を参照してください。ただし、以下の部分は置換えの対象にはなりません。

- ・ 注釈
- ・ コメント記述文字によるコメント
- ・ 要約定義によるコメント
- ・ 文字列リテラル
- ・ 目的言語記述
- ・ システム定義のキーワード

置換えの手順

文字列の置換えは以下の手順で行われます。

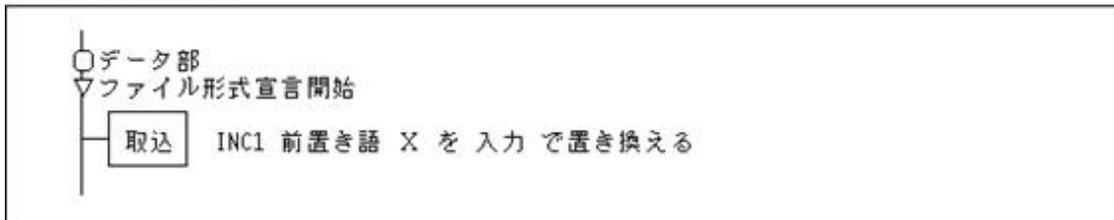
1. 本文中の、以下の文字で区切られる文字列が、置換えの対象になります。
 - 全角または半角の空白
 - 全角または半角の特殊文字
 - システム定義のキーワード

ただし、「(」と「)」は、区切り文字であると同時に、置換えの対象にもなります。

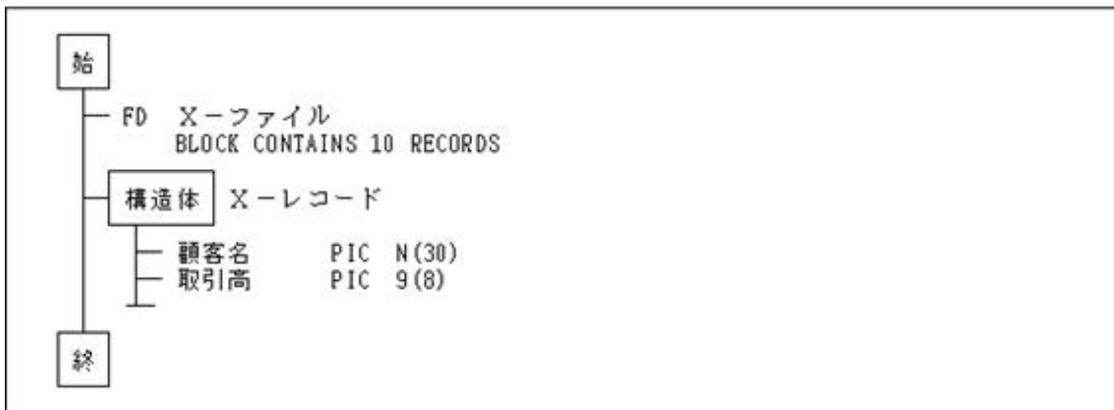
- [形式1]の場合、手順1で得られた置換え対象の文字列が語1に一致していたならば、対応する語2に置換えられます。
この一致の比較は、語1を指定した順に行われます。複数の語1を指定し、かつ途中で一致した場合には、残りの語1との比較は行われません。
- [形式2]の場合、手順1で得られた置換え対象の文字列の左端が語3の右側に全角または半角の「-」を繋げたもの(システム定義のキーワード「前置き語」を指定した場合)か、または、置換え対象の文字列の右端が語3の左側に全角または半角の「-」を繋げたもの(システム定義のキーワード「後置き語」を指定した場合)ならば、対応する語4に置換えられます。
この一致の比較は、語3を指定した順に行われます。複数の語3を指定し、かつ途中で一致した場合には、残りの語3との比較は行われません。ただし、前置き語が置換えられた文字列でも、後置き語の置換え対象にはなりません。その逆も同じです。
- すべての置換え対象の文字列が評価されるまで、手順1～3が繰り返されます。

記述例

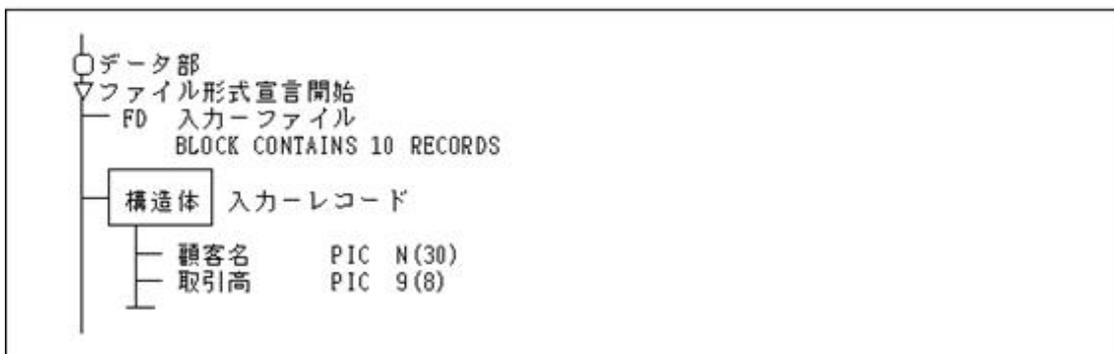
【プログラム仕様書】



【インクルード仕様書 (INC1)】



↓ 取り込んだ結果



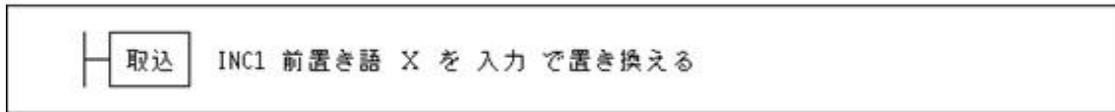
注意

一般規則で示したように、構文定義機能のキーワード(利用者定義のキーワード)は、置換え対象を区切る文字列ではありません。よって、構文定義機能の引用文または引用式のパラメタを置換え対象にする場合には、パラメタとキーワードの間に区切り文字の空白を記述する必要があります。

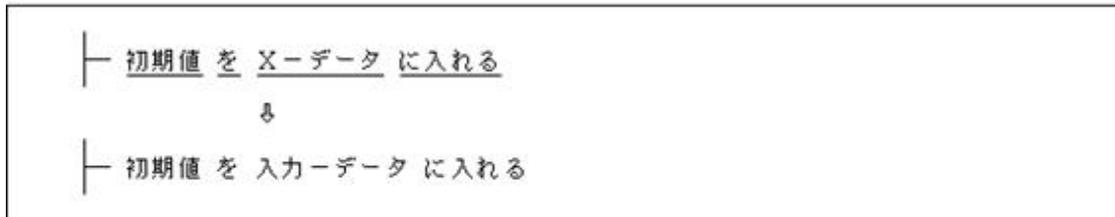
正しい記述例と誤った記述例を“[図5.2 構文定義機能の引用文のパラメタを置き換える場合の記述例](#)”に示します。インクルード仕様書中の下線で示した文字列が、置換えの対象となります。

図5.2 構文定義機能の引用文のパラメタを置き換える場合の記述例

〔取込み表記の記述〕



〔正しいインクルード仕様書の記述例と置換え結果〕



〔誤ったインクルード仕様書の記述例と置換え結果〕



正しい例では、「X-データ」の前後を空白で区切っているため、「X-データ」が置換え対象の文字列とみなされ、「入力-データ」に置換えられます。

誤った例では、利用者定義のキーワード「を」と「に入れる」は、置換え対象を区切る文字列ではないため、「初期値をX-データに入れる」が置換え対象の文字列とみなされ、意図した置換は行われません。

5.4 インクルード仕様書

インクルード仕様書は、取込み表記によって取り込まれる仕様書です。

構成



記述規則

インクルード仕様書は、プログラム仕様書と同様にモジュール関連情報とモジュール本体から構成されます。モジュール関連情報については“[3.1.1 モジュール関連情報](#)”を参照してください。

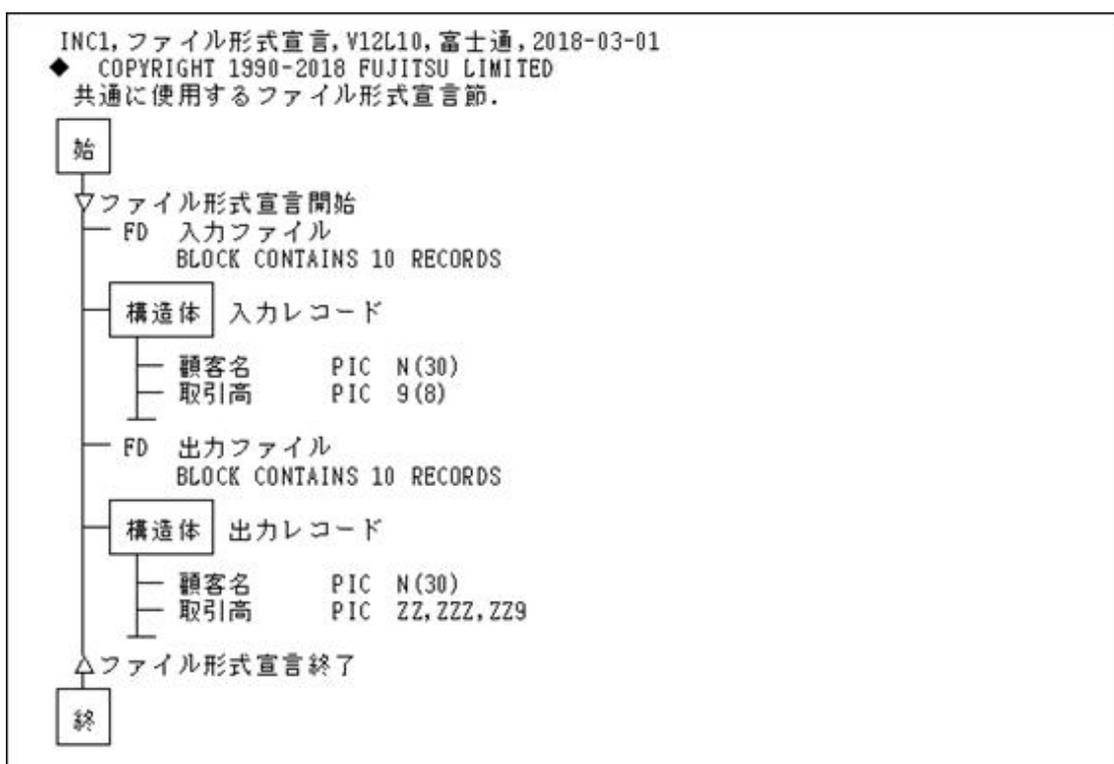
インクルード仕様書のモジュール本体には、取り込む表記を記述します。以下に示す表記を記述することができます。

- ・ 条件付き翻訳判定、条件付き翻訳単一選択
- ・ 取込み
- ・ コメントキーワード定義文
- ・ 抽象化処理構文定義、一時変数宣言文、抽象化条件式定義文、引用文
- ・ 日本語名標宣言文
- ・ 内部プログラム単位
- ・ 環境部を構成する表記、環境部の節を構成する表記
- ・ プログラム内データ部を構成する表記、データ部の節を構成する表記
- ・ 手続き部を構成する表記、手続き部の節を構成する表記
- ・ 部および節に依存しない表記

ただし、以下の表記を記述する場合には、開始と終了を対応させて記述してください。

- ・ 抽象化処理構文定義
- ・ 内部プログラム単位
- ・ 環境部、プログラム内データ部および手続き部の節
- ・ 宣言部分

記述例



第6章 コメントキーワード定義機能

本章では、コメントキーワード定義機能について説明します。

6.1 コメントキーワード定義機能とは

コメントキーワードとは、そのキーワードの文字列より前または後ろの文字列がコメントであることを表す文字列です。コメントキーワードを用いることにより、本文中の処理や条件に自然な補足説明(コメント)を付加することができます。

コメントキーワード定義機能とは、このコメントキーワードを定義する機能です。

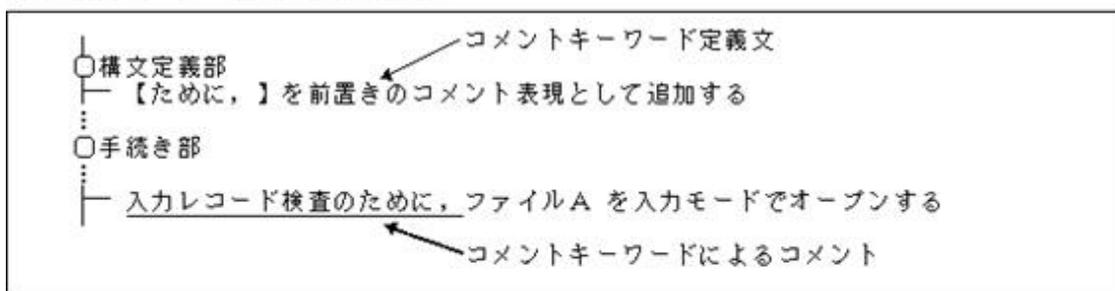
コメント記述文字によるコメントとコメントキーワードによるコメントを“[図6.1 コメント記述文字によるコメントとコメントキーワードによるコメント](#)”に示します。

図6.1 コメント記述文字によるコメントとコメントキーワードによるコメント

【コメント記述文字によるコメント】

```
| ファイルA を入力モードでオープンする ☆入力レコード検査のため☆
```

【コメントキーワードによるコメント】



前置きコメントキーワードと後置きコメントキーワード

コメントキーワードには、前置きコメントキーワードと後置きコメントキーワードがあります。

前置きコメントキーワードは、そのキーワードよりも前の文字列がコメントであることを表します。“[図6.1 コメント記述文字によるコメントとコメントキーワードによるコメント](#)”の例では、前置きコメントキーワードを使用しています。

後置きコメントキーワードは、そのキーワードよりも後の文字列がコメントであることを表します。

標準コメントキーワードと利用者定義のコメントキーワード

前置きコメントキーワードおよび後置きコメントキーワードには、それぞれ標準コメントキーワードと利用者定義のコメントキーワードがあります。

標準コメントキーワードは、システムがあらかじめ用意したコメントキーワードで、定義せずにそのまま使用することができます。標準コメントキーワードとしてどのようなものが用意されているかは、“[6.3 標準コメントキーワード](#)”を参照してください。

利用者定義のコメントキーワードは、利用者がコメントキーワード定義文により定義するコメントキーワードです。定義方法については“[6.2 コメントキーワード定義文](#)”を参照してください。

コメントキーワードの引用

コメントキーワードを用いてコメントを付加することを、コメントキーワードを引用するといいます。コメントキーワードを引用できる表記には制限があり、どの表記の本文でも引用できる訳ではありません。詳細は“[6.4 コメントキーワードの引用](#)”を参照してください。

また、文字列リテラル内にコメントキーワードと同じ文字列が存在しても、コメントキーワードの引用とはみなされません。

6.2 コメントキーワード定義文

コメントキーワード定義文は、利用者定義のコメントキーワードを定義する文です。

記述形式

[前置きコメントキーワード定義文]

┆ 【コメントキーワード】 を前置きのコメント表現として追加する

[後置きコメントキーワード定義文]

┆ 【コメントキーワード】 を後置きのコメント表現として追加する

記述規則

コメントキーワード定義文は、構文定義部に記述します。ただし、抽象化処理構文定義の定義体内には記述することはできません。

コメントキーワードには、1文字以上30文字以下の全角または半角の文字列を指定します。文字列には、全角および半角の空白を含めることはできません。

記述例

○ 構文定義部
┆ 【ために、】を前置きのコメント表現として追加する
┆ 【になったら】を後置きのコメント表現として追加する

6.3 標準コメントキーワード

標準コメントキーワードは、システムがあらかじめ用意したコメントキーワードです。コメントキーワード定義文で定義しないで使用することができます。

標準コメントキーワードには、前置きコメントキーワードと後置きコメントキーワードがあります。

標準の前置きコメントキーワード

標準コメントキーワードとして用意されている前置きコメントキーワードを以下に示します。

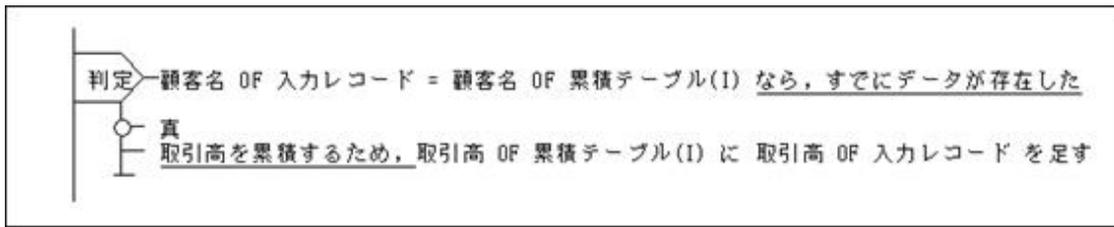
- ・ [コメント]ため、

標準の後置きコメントキーワード

標準コメントキーワードとして用意されている後置きコメントキーワードを以下に示します。

- ・ なら[コメント]
- ・ を調べ[コメント]
- ・ の間[コメント]
- ・ になるまで[コメント]

引用例



備考: 下線部分がコメントとみなされます。

6.4 コメントキーワードの引用

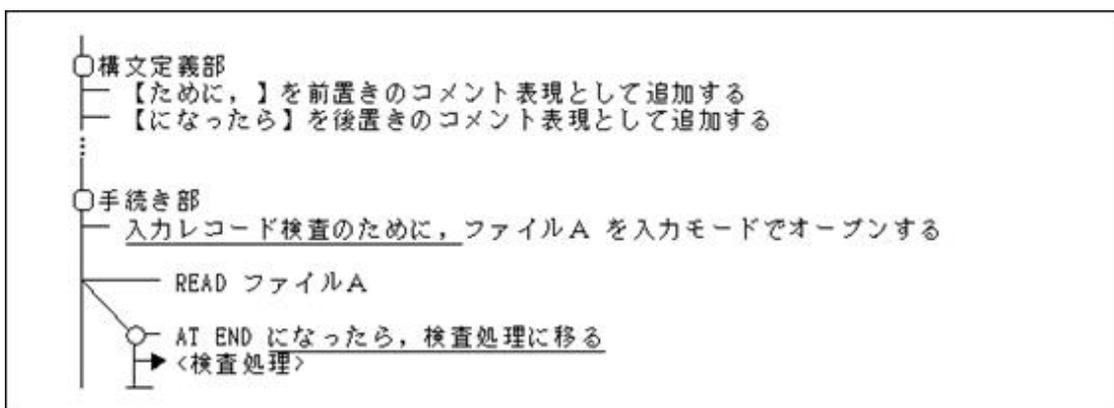
コメントキーワードを用いてコメントを付加することを、コメントキーワードを引用するといいます。

引用できる場所

コメントキーワードは、コメントキーワード定義文を記述した翻訳単位の以下の表記の本文で引用することができます。

- ・ 抽象化処理構文の引用文
- ・ 宣言文、順次文、空文
空文は後置きコメントキーワードしか引用できません。
- ・ 選択系の表記(条件付き翻訳は含みません)
- ・ 選択節(条件付き翻訳に従属する選択節は含みません)
本文がシステム定義のキーワード(「真」、「偽」、「その他」および「検索条件」)の場合には、後置きコメントキーワードしか引用できません。
- ・ ループ系の表記
本文がシステム定義のキーワード(「*」)の場合、またはループ制御(変数型、更新型、指定回数型)の場合には、後置きコメントキーワードしか引用できません。
- ・ 例外処理、検索処理
ただし、文字列リテラル内にコメントキーワードと同じ文字列が存在しても、コメントキーワードの引用とはみなされません。

引用例



備考: 下線部分がコメントとみなされます。

対象から除外する方法

コメントキーワードの引用以外の目的で、コメントキーワードと同じ文字列を使用したい場合には、構文定義のエスケープ文字で文字列を囲みます。詳細は“7.4 構文定義のエスケープ文字”を参照してください。

第7章 構文定義機能

本章では、構文定義機能について説明します。

7.1 構文定義機能とは

構文定義機能とは、一連の処理および条件を一文の日本語で記述するための機能です。

構文定義機能は、以下の目的で利用できます。

- COBOLの文および条件式を日本語化する。
- 複雑な手続きを一文で記述する。
- 使用頻度の高い手続きを部品化/標準化する。

構文定義機能の種類

構文定義機能には、以下の二つがあります。

抽象化処理構文定義機能:

一連の処理または宣言に対して日本語の構文を定義します。定義した構文を抽象化処理構文といいます。

抽象化条件式定義機能:

COBOLの条件式に対して日本語の構文を定義します。定義した構文を抽象化条件式構文といいます。

これらの機能を利用して利用者が定義した構文を**利用者定義構文**といいます。また、利用者が定義しなくても利用できるようにシステムがあらかじめ用意した構文を**標準構文**といいます。

内部利用者定義構文と外部利用者定義構文

利用者定義構文は、プログラム仕様書または外部構文定義仕様書の構文定義部で定義します。プログラム仕様書の構文定義部で定義した利用者定義構文を**内部利用者定義構文**、外部構文定義仕様書の構文定義部で定義した利用者定義構文を**外部利用者定義構文**といいます。

標準構文の種類

標準構文には、以下の3つがあります。

YPS標準構文:

YPSの各言語に共通する標準構文です。

YPS/COBOL標準構文:

YPS/COBOL言語の標準構文です。

YPS/COBOL事務処理構文:

事務処理に関する構文の集まりです。YPS/COBOL事務処理構文は、標準日本語名標と組み合わせて使用します。

各標準構文にどのような構文が用意されているかは、“[付録B 標準構文](#)”を参照してください。

構文の引用

構文定義機能で定義した構文(標準構文を含む)を使用することを、構文を引用するといいます。詳細については“[7.2.4 抽象化処理構文の引用](#)”および“[7.3.4 抽象化条件式構文の引用](#)”を参照してください。

7.2 抽象化処理構文定義機能

抽象化処理構文定義機能とは、一連の処理または一連の宣言に対して日本語の構文を定義する機能です。

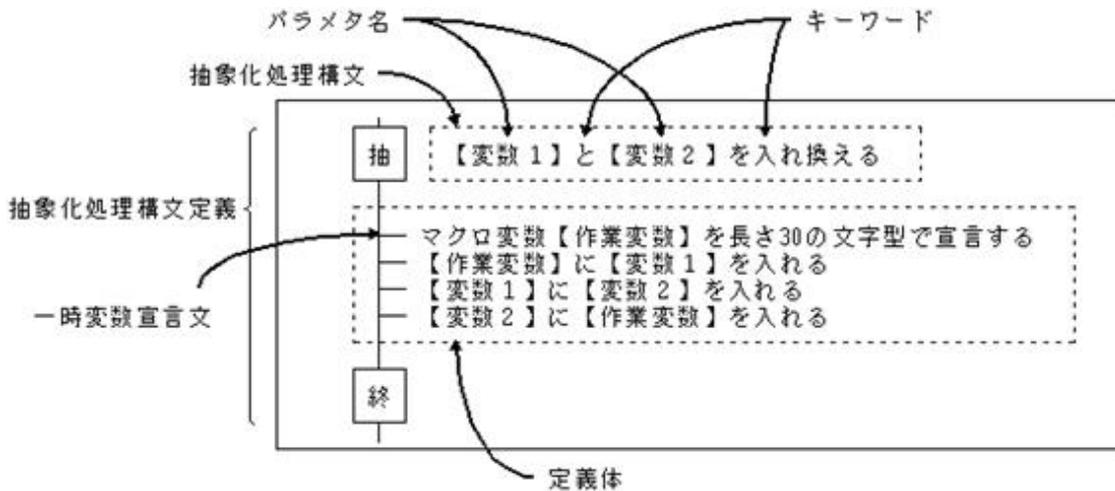
構文を定義する表記を**抽象化処理構文定義**といいます。

定義する一連の処理または一連の宣言を**定義体**、構文を**抽象化処理構文**といいます。

定義体の中だけで使用する変数を**一時変数**、一時変数を宣言する文を**一時変数宣言文**といいます。

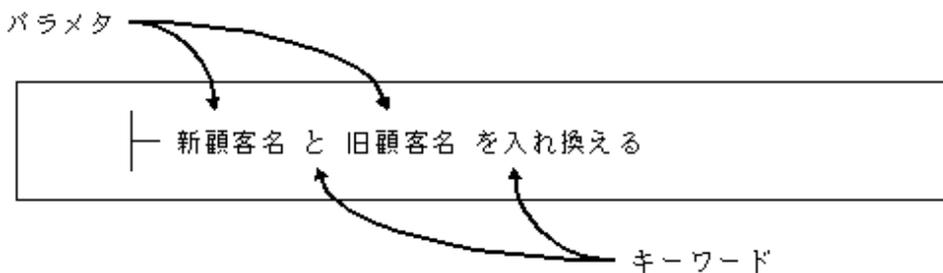
抽象化処理構文の定義例を“[図7.1 抽象化処理構文の定義例](#)”に、抽象化処理構文の引用例を“[図7.2 抽象化処理構文の引用例](#)”に、それぞれ示します。

図7.1 抽象化処理構文の定義例



定義体内でパラメタおよび一時変数を参照する場合には、システム定義のキーワード「【】と【】」で囲んで、パラメタ名または一時変数名を記述します。

図7.2 抽象化処理構文の引用例



抽象化処理構文を引用する場合には、引用する抽象化処理構文のキーワードと同じ文字列を同じ順番で記述し、パラメタを記述すべきところにパラメタとなる文字列を記述します。

7.2.1 抽象化処理構文定義

抽象化処理構文定義は、抽象化処理構文を定義します。

記述形式



記述規則

抽象化処理構文定義は、プログラム仕様書または外部構文定義仕様書の構文定義部に記述します。ただし、抽象化処理構文の定義体内に記述することはできません。

抽象化処理構文定義開始図記号の本文には、抽象化処理構文を記述します。抽象化処理構文については“7.2.2 抽象化処理構文”を参照してください。

定義体

定義体は、抽象化処理構文として定義する、一連の処理または一連の宣言を記述するところです。定義体の先頭には、一時変数宣言文を記述します。複数の一時変数を宣言する場合には、連続して記述してください。定義体で一時変数を使用しなければ、一時変数宣言文は省略することができます。一時変数宣言文については“7.2.3 一時変数宣言文”を参照してください。

定義体内には、一時変数宣言文以外に、以下の表記を記述することができます。

- 条件付き翻訳判定、条件付き翻訳単一選択
- 取込み
- 引用文
- 環境部の節を構成する表記
- データ部の節を構成する表記
- 手続き部の節を構成する表記
- 部および節に依存しない表記

定義体内でパラメタおよび一時変数を参照する場合には、システム定義のキーワード「【】と「」」で囲んで、パラメタ名または一時変数名を記述します。

注意事項

抽象化処理構文定義の定義体内には、その抽象化処理構文を引用する場所に記述できる表記を記述してください。

7.2.2 抽象化処理構文

抽象化処理構文とは、定義体として記述した一連の処理または一連の宣言に対して定義する構文です。

抽象化処理構文は、構文を一意に識別するための文字列である**キーワード**と、構文を引用する場所によって変更することができる**パラメタ**から構成されます。

記述形式

`{ [[パラメタ名]] キーワード } [[パラメタ名]]`

記述規則

抽象化処理構文には、1つ以上のキーワードと0個以上のパラメタを記述します。キーワードを連続して記述することはできませんが、パラメタを連続して記述することはできません。キーワードを連続して記述する場合には、1つ以上の全角または半角の空白で区切ってください。

キーワード

キーワードには、1文字以上の全角または半角の文字列を記述します。文字列には、空白、「「」、「」」、「《》および「》」を含めることはできません。

パラメタ

パラメタは、パラメタ名をシステム定義のキーワード「【】と「」」で囲みます。

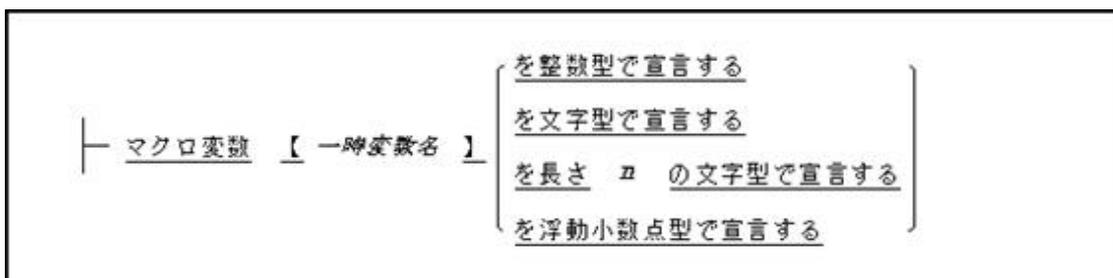
パラメタ名には、1文字以上30文字以下の全角または半角の文字列を記述します。文字列には空白を含めることはできません。

パラメタ名は、抽象化処理構文内で一意でなければなりません。

7.2.3 一時変数宣言文

一時変数宣言文は、定義体の中だけで使用する変数(**一時変数**)を宣言する文です。

記述形式



記述規則

一時変数宣言文は、定義体の先頭に連続して記述します。

一時変数名には、1文字以上30文字以下の全角または半角の文字列を記述します。文字列には空白を含めることはできません。

n には、1以上32767以下の整数を指定します。全角でも半角でも記述することができます。

一般規則

一時変数のデータ型とCOBOLのデータの型との対応を“表7.1 一時変数とCOBOLのデータ型の対応”に示します。

表7.1 一時変数とCOBOLのデータ型の対応

一時変数のデータ型	COBOLのデータ型
整数型	PIC S9(8) BINARY
文字型	PIC X(1)
長さ n の文字型	PIC X(n)
浮動小数点型	COMP-1

7.2.4 抽象化処理構文の引用

抽象化処理構文定義で定義した抽象化処理構文(標準構文を含む)を使用することを、抽象化処理構文を引用するといいます。

抽象化処理構文の引用は、**抽象化処理構文の引用文**で行います。抽象化処理構文の引用文のことを単に**引用文**ということもあります。

記述形式

【形式1】



【形式2】



記述規則

抽象化処理構文の引用文は、以下の場所に記述することができます。

- ・ [形式1]の場合
 - 抽象化処理構文定義の定義体
 - 環境部の各節

図7.4 同じ抽象化処理構文が存在した場合の優先順位

順序	構文の種類	抽象化処理構文定義の記述場所
高 ↓ 低	内部利用者定義構文	構文定義部の末尾 ↓ 構文定義部の先頭
	利用者定義構文	オプションで最後に指定された外部構文定義仕様書 ↓ オプションで最初に指定された外部構文定義仕様書
	外部利用者定義構文	構文定義部の末尾 ↓ 構文定義部の先頭 ↓ 構文定義部の末尾 ↓ 構文定義部の先頭
	標準構文	_____

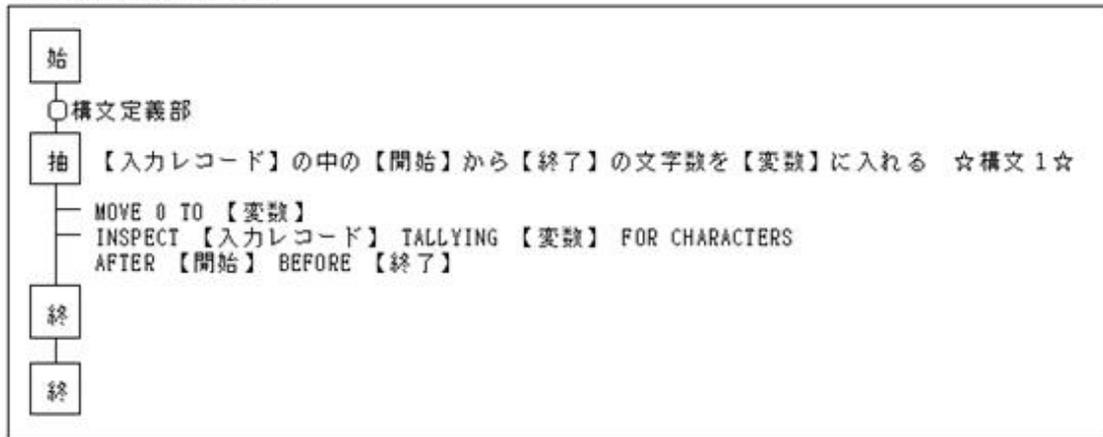
引用文に一致する抽象化処理構文の検索方法の例

引用文に一致する抽象化処理構文の検索方法の例を“[図7.5 引用文に一致する抽象化処理構文の検索方法の例](#)”に示します。

図7.5 引用文に一致する抽象化処理構文の検索方法の例

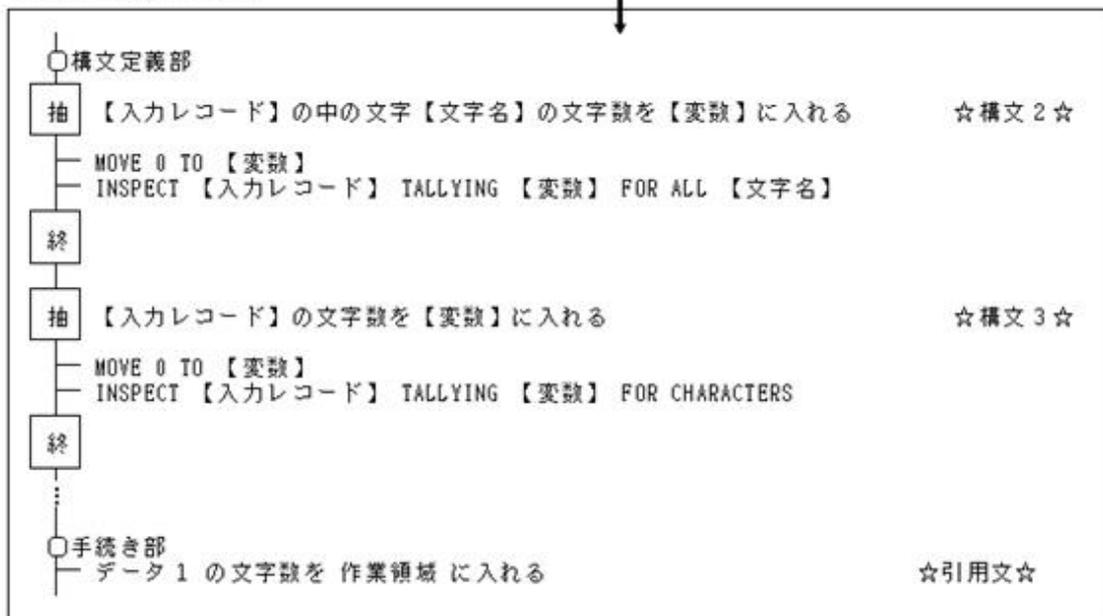
条件： プログラム仕様書Aの翻訳時に、外部構文定義仕様書Bを指定する

【外部構文定義仕様書B】



【プログラム仕様書A】

翻訳時に指定



[説明]

1. 抽象化処理構文の最右端のパラメタが現れるまでのキーワード群の文字数を右端から数えます。
 構文1: “に入れる”の4文字
 構文2: “に入れる”の4文字
 構文3: “に入れる”の4文字
 →キーワード群の文字数が同じなので順序は決まりません。
2. 次のキーワード群の文字数を数えます。
 構文1: “の文字数を”の5文字
 構文2: “の文字数を”の5文字
 構文3: “の文字数を”の5文字
 →キーワード群の文字数が同じなので順序は決まりません。

3. 次のキーワード群の文字数を数えます。

構文1:“から”の2文字

構文2:“の中の文字”の5文字

構文3:0文字

→キーワード群の文字数により、以下の順番で検索されます。

構文2→構文1→構文3

7.3 抽象化条件式定義機能

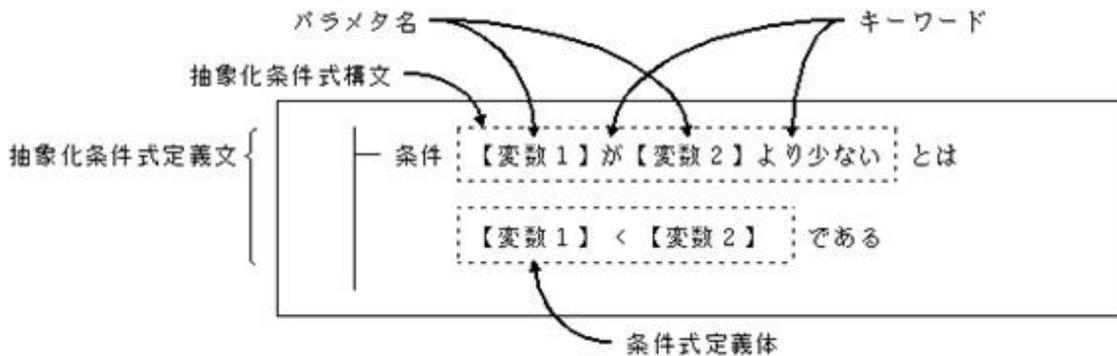
抽象化条件式定義機能とは、COBOLの条件式に対して日本語の構文を定義する機能です。

構文を定義する表記を抽象化条件式定義文といいます。

定義する条件式を条件式定義体、構文を抽象化条件式構文といいます。

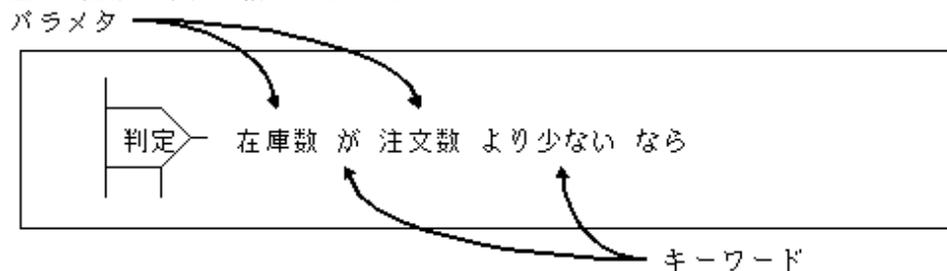
抽象化条件式構文の定義例を“[図7.6 抽象化条件式構文の定義例](#)”に、抽象化条件式構文の引用例を“[図7.7 抽象化条件式構文の引用例](#)”に、それぞれ示します。

図7.6 抽象化条件式構文の定義例



条件式定義体でパラメタを参照する場合には、システム定義のキーワード「【と】」で囲んでパラメタ名を記述します。

図7.7 抽象化条件式構文の引用例



抽象化条件式構文を引用する場合には、引用する抽象化条件式構文のキーワードと同じ文字列を同じ順番で記述し、パラメタを記述すべきところにパラメタとなる文字列を記述します。

7.3.1 抽象化条件式定義文

抽象化条件式定義文は、抽象化条件式構文を定義します。

記述形式

— 条件 抽象化条件式構文 とは 条件式定義体 である

記述規則

抽象化条件式定義文は、プログラム仕様書または外部構文定義仕様書の構文定義部に記述します。ただし、抽象化処理構文定義の定義体内に記述することはできません。

抽象化条件式構文については“7.3.2 抽象化条件式構文”を参照してください。

条件式定義体

条件式定義体は、抽象化条件式構文として定義する条件式です。

条件式定義体には、条件式を記述します。記述規則については“7.3.3 条件式”を参照してください。

条件式定義体でパラメタを参照する場合には、システム定義のキーワード「【】と「」」で囲んでパラメタ名を記述します。

7.3.2 抽象化条件式構文

抽象化条件式構文とは、ある条件式に対して定義する日本語の条件式です。

抽象化条件式構文は、構文を一意に識別するための文字列である**キーワード**と、構文を引用する場所によって変更することができる**パラメタ**から構成されます。

記述形式

```
{ [ [ パラメタ名 ] キーワード ] [ [ パラメタ名 ] ] }
```

記述規則

抽象化条件式構文には、1つ以上のキーワードと0個以上のパラメタを記述します。キーワードは連続して記述することができますが、パラメタは連続して記述することはできません。キーワードを連続して記述する場合には、1つ以上の全角または半角の空白で区切ってください。

キーワード

キーワードには、1文字以上の全角または半角の文字列を記述します。文字列には空白、「(、)」、「[、]」、「《》および「》」を含めることはできません。

パラメタ

パラメタは、パラメタ名をシステム定義のキーワード「【】と「」」で囲みます。

パラメタ名には、1文字以上30文字以下の全角または半角の文字列を記述します。文字列には空白を含めることはできません。

パラメタ名は、抽象化条件式構文内で一意でなければなりません。

7.3.3 条件式

条件式は、評価の結果として真または偽の論理値を返す式です。

記述形式

[基本形]

```
{ 抽象化条件式構文の引用式  
  COBOLの条件式 }
```

[応用形1(日本語論理演算子による結合)]

条件式 日本語論理演算子 条件式

[応用形2(解釈および結合順序の変更)]

(条件式)

記述規則

条件式には、抽象化条件式の引用式またはCOBOLの条件式を記述します。抽象化条件式の引用式については“[7.3.4 抽象化条件式構文の引用](#)”を参照してください。

日本語論理演算子による結合

条件式を、日本語論理演算子を用いて結合することができます。

日本語論理演算子には、以下の2つがあります。

- かつ
- または

解釈および結合順序の変更

条件式をCOBOLの分離符「(」と「)」で囲み、解釈および結合の順序を変更することができます。

記述例

[基本形]

A が B と等しい
X = Y OR Z

[応用形1(日本語論理演算子による結合)]

A が B と等しい または X = Y OR Z

[応用形2(解釈および評価順序の変更)]

(A が B と等しい または X = Y OR Z) でない

備考:「～が～と等しい」および「～でない」という抽象化条件式構文が定義されているものと仮定します。

一般規則

抽象化条件式構文の引用式は、引用の結果として展開されるCOBOLの条件式と等価です。

日本語論理演算子による結合

日本語論理演算子「かつ」および「または」は、それぞれ以下に示すCOBOLの論理演算子と等価です。

- かつ:
AND
- または:
OR

解釈および結合順序の変更

解釈および結合の順序を変更するCOBOLの分離符「(」と「)」で囲んだ条件式は、「(」と「)」で囲んでいない条件式より先に解釈および結合されます。ここで、解釈されるとは、YPS仕様書の翻訳時に、抽象化条件式の引用式またはCOBOLの条件式とみなされることを表しています。



COBOLの用語

条件式

論理演算子

7.3.4 抽象化条件式構文の引用

抽象化条件式定義文で定義した抽象化条件式構文(標準構文を含む)を使用することを、抽象化条件式構文を引用するといいます。抽象化条件式構文の引用は、**抽象化条件式構文の引用式**で行います。抽象化条件式の引用式のことを単に**引用式**ということもあります。

記述規則

抽象化条件式の引用式は、条件式という記述形式を持つYPS表記の本文に記述することができます。抽象化条件式定義文の条件式定義体に記述することもできます。

抽象化条件式構文の引用式は、引用する抽象化条件式構文のキーワードと同じ文字列を同じ順序で記述し、パラメタを記述すべきところにパラメタとなる文字列を記述します。パラメタとキーワードの間、およびキーワードとキーワードの間には、全角または半角の空白を入れることもできます。

また、抽象化条件式定義文の条件式定義体には、直接または間接的に自分自身を引用する引用式を記述することはできません。

一般規則

抽象化条件式構文の引用の一般規則は、抽象化処理構文の引用の一般規則と同じです。“7.2.4 抽象化処理構文の引用”を参照してください。その際、用語を以下のように置き換えてください。

- 引用文 → 引用式
- 抽象化処理構文 → 抽象化条件式構文
- 抽象化処理構文定義 → 抽象化条件式定義文

7.4 構文定義のエスケープ文字

構文定義のエスケープ文字は、以下の場合に使用します。

- コメントキーワードと同じ文字列を、コメントキーワード定義機能の作用対象から除外する。
- 構文定義機能のキーワードと同じ文字列を、構文定義機能の作用対象から除外する。

記述形式

`「 除外する文字列 」`

記述規則

コメントキーワード定義機能および構文定義機能の処理対象から除外したい文字列を、システム定義のキーワード「(」と「)」で囲みます。システム定義のキーワード「(」と「)」は全角で記述します。「(」と「)」は、1つの表記内で対応させて記述してください。

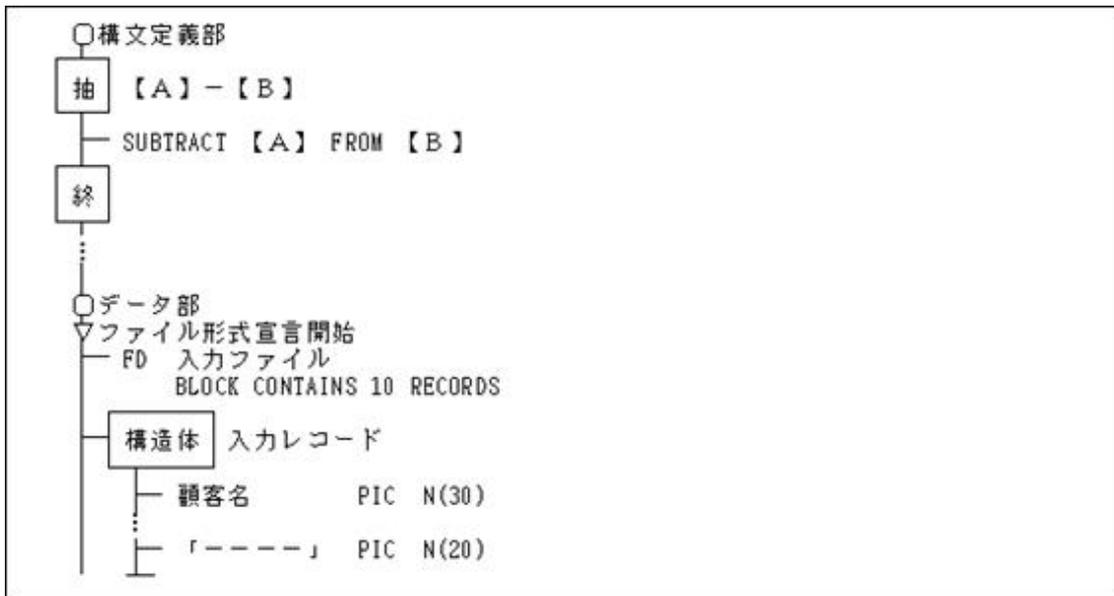
構文定義のエスケープ文字は入れ子にすることはできません。

一般規則

構文定義のエスケープ文字で囲った文字列は、コメントキーワード定義機能および構文定義機能の作用対象から除外されます。
以下に示す場所は、YPS機能の作用対象外であるため、「|」と「|」は構文定義のエスケープ文字とみなされません。

- ・ 文字列リテラル
- ・ 目的言語記述

記述例



7.5 外部構文定義仕様書

外部構文定義仕様書は、複数のYPS仕様書で引用するコメントキーワード、抽象化処理構文および抽象化条件式構文を定義する仕様書です。

プログラム仕様書の翻訳時にオプションを指定することにより、外部構文定義仕様書で定義したコメントキーワード、抽象化処理構文および抽象化条件式構文が引用できるようになります。

構成



記述規則

外部構文定義仕様書は、プログラム仕様書と同様にモジュール関連情報とモジュール本体から構成されます。モジュール関連情報については“3.1.1 モジュール関連情報”を参照してください。

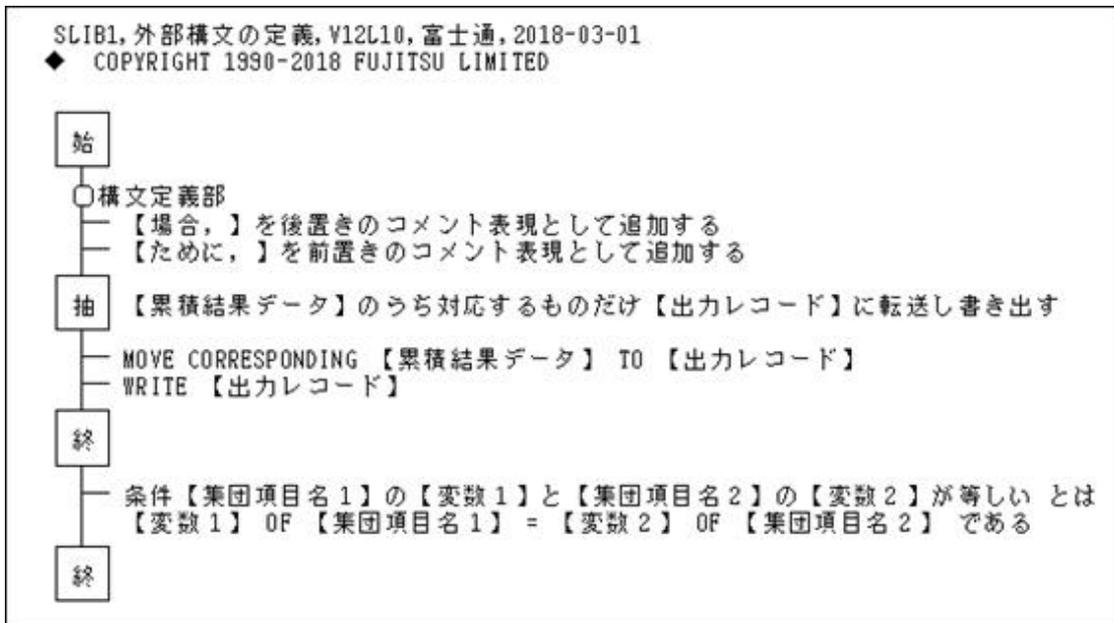
モジュール本体開始の直後には、構文定義部開始を記述します。

外部構文定義仕様書の構文定義部には、コメントキーワード定義文、抽象化処理構文定義および抽象化条件式定義文を記述します。

一般規則

外部構文定義仕様書に記述したコメントキーワード定義文、抽象化処理構文定義および抽象化条件式定義文は、プログラム仕様書の構文定義部の先頭に記述されたのと同等に扱われます。

記述例



第8章 日本語名標宣言機能

本章では、日本語名標宣言機能について説明します。

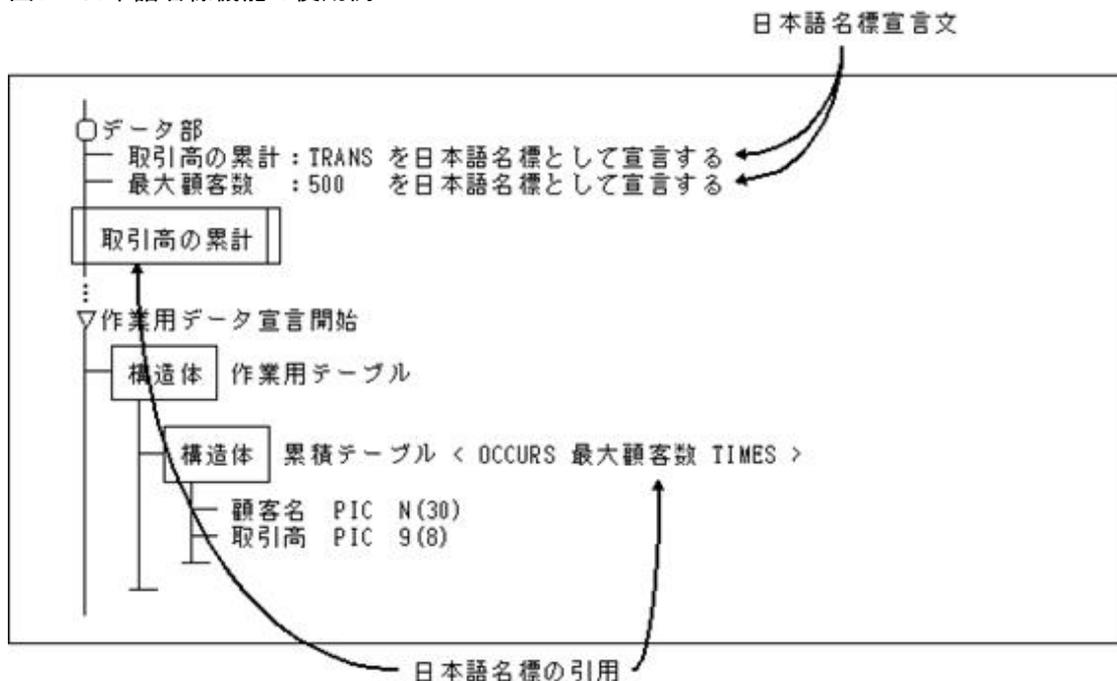
8.1 日本語名標宣言機能とは

日本語名標宣言機能とは、利用者がYPS仕様書の本文に記述するCOBOLの文字列に対して、日本語の名標を付けるための、名標置換え機能です。

日本語名標宣言文で、利用者がYPS仕様書内で使用するCOBOLの文字列に対する日本語の名標を宣言することにより、以降の仕様書の本文で日本語の名標が使用できるようになります。

“[図8.1 日本語名標機能の使用例](#)”に日本語名標宣言機能の使用例を示します。使用例では、プログラム名と数字定数を日本語名標として宣言しています。

図8.1 日本語名標機能の使用例



8.2 日本語名標宣言文

日本語名標宣言文は、日本語名標を宣言する文です。

記述形式

```
— 文字列1 [ : 文字列2 ] を日本語名標として宣言する
```

記述規則

日本語名標宣言文は、モジュール共通データ部およびプログラム内データ部に記述します。

プログラム内データ部に記述する場合には、プログラム内データ部を構成する節内に記述します。構造体内に記述することもできます。

文字列1と文字列2を区切るシステム定義のキーワード「:」は全角でも半角でも記述できます。

文字列2は省略することができます。

文字列1

文字列1には、仕様書内で使用する日本語の名標を記述します。

日本語の名標は、以下のいずれかの文字列を指定します。

- 空白および特殊文字を含まない1文字以上30文字以下の全角の文字列
- 1文字の全角の特殊文字

特殊文字

ここでいう**特殊文字**とは、以下の文字を指しています。

+(正符号)	*(アスタリスク)	/ (斜線)	= (等号)	¥ (円記号)	, (カンマ)
.(ピリオド)	" (二重引用符)	((始め丸括弧)) (終わり丸括弧)	> (不等号より大)	
< (不等号より小)	: (コロン)	% (パーセント)	_ (アンダーライン)	? (疑問符)	
' (引用符)	@ (単位記号)	# (井げた)	\$ (ドル記号)		

文字列2

文字列2には、翻訳時に置き換えるCOBOLの文字列を記述します。

一般規則

日本語名標宣言機能の置換え対象は、日本語名標宣言文を記述した次の表記から、翻訳単位の終わりまたは文字列1が再び宣言されるまでの、表記の本文です。ただし、以下の部分は置換えの対象にはなりません。

- ・ 注釈
- ・ 文字列リテラル
- ・ 目的言語記述
- ・ YPSの文字列

日本語名標の引用以外の目的で、文字列1と同じ文字列を使用したい場合には、日本語記述のエスケープ文字で文字列を囲みます。詳細は“[8.3 日本語記述のエスケープ文字](#)”を参照してください。

文字列2を省略した場合には、YPSコンパイラにより翻訳単位内で一意になる名標が自動生成されます。

置換えの手順

日本語名標の置換えは以下の手順で行われます。

1. 本文中の、以下の文字で区切られる文字列が、置換えの対象になります。

- 半角文字
- 全角の空白
- 全角の特殊文字
- システム定義のキーワード

ただし、全角の特殊文字は、区切り文字であると同時に、置換えの対象にもなります。

2. 手順1で得られた置換え対象の文字列が文字列1に一致していたならば、対応する文字列2に置換えられます。

この一致の比較は、日本語名標宣言文を記述した順に行われます。複数の日本語名標宣言文を記述し、かつ途中で一致した場合には、残りの文字列1との比較は行われません。

3. すべての置換え対象の文字列が評価されるまで、手順1および2が繰り返されます。

記述例



8.3 日本語記述のエスケープ文字

日本語記述のエスケープ文字は、以下の場合に使用します。

- 日本語名標宣言文の文字列1と同じ文字列を、日本語名標宣言機能の作用対象から除外する。
- 全角文字記述機能の処理対象となる文字列を、全角文字記述機能の作用対象から除外する。

記述形式

《 除外する文字列 》

記述規則

日本語名標宣言機能および全角文字記述機能の処理対象から除外したい文字列を、システム定義のキーワード「《と『』」で囲みます。システム定義のキーワード「《と『』」は全角で記述します。「《と『』」は、一つの表記内で対応させて記述してください。

日本語記述のエスケープ文字は入れ子にすることはできません。

一般規則

日本語記述のエスケープ文字で囲った文字列は、日本語名標宣言機能および全角文字記述機能の作用対象から除外されます。

以下に示す場所は、YPS機能の作用対象外であるため、「《と『』」は日本語記述のエスケープ文字とはみなされません。

- 文字列リテラル内
- 目的言語記述内

8.4 外部日本語名標宣言仕様書

外部日本語名標宣言仕様書は、複数のYPS仕様書で引用する日本語名標を宣言する仕様書です。

プログラム仕様書の翻訳時にオプションを指定することにより、外部日本語名標宣言仕様書で宣言した日本語名標が引用できるようになります。

構成



記述規則

外部日本語名標宣言仕様書は、プログラム仕様書と同様にモジュール関連情報とモジュール本体から構成されます。モジュール関連情報については“[3.1.1 モジュール関連情報](#)”を参照してください。

モジュール本体開始の直後には、データ部開始を記述します。

日本語名標宣言仕様書のデータ部には、1つ以上の日本語名標宣言文を記述します。

一般規則

外部日本語名標宣言仕様書に記述した日本語名標宣言文は、プログラム仕様書のモジュール共通データ部の先頭に記述されたのと同等に扱われます。

記述例



第9章 全角文字記述機能

本章では、全角文字記述機能について説明します。

9.1 全角文字記述機能とは

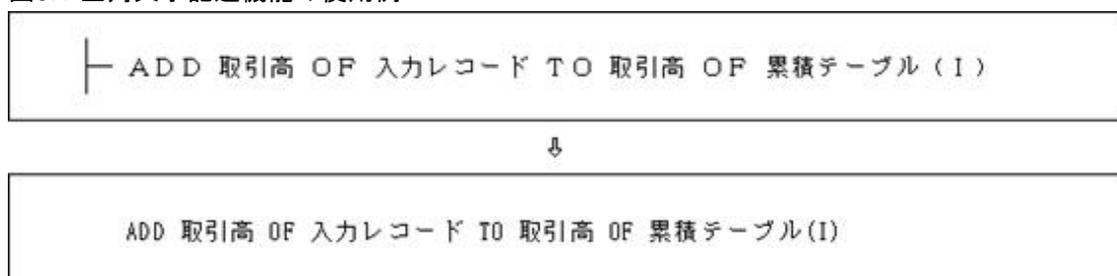
全角文字記述機能とは、本文中の全角の文字列を半角の文字列へ変換する機能です。

全角文字記述機能を使用することにより、半角で記述しなければならないCOBOLの文字列を全角で記述できるようになります。

全角文字記述機能を使用する場合には、翻訳指示行に「全角文字記述(あり)」を指定するか、翻訳時にオプションを指定する必要があります。翻訳オプションの指定方法については、各システムの使用手引書を参照してください。

“[図9.1 全角文字記述機能の使用例](#)”に全角文字記述機能の使用例を示します。使用例では、COBOL予約語「ADD」、「OF」および「TO」を全角で記述しています。

図9.1 全角文字記述機能の使用例



9.2 全角文字記述機能の対象となる文字列

記述規則

本文中に半角で記述するCOBOLの文字列を全角で記述します。ただし、文字列リテラル内および目的言語記述内は当機能の作用対象外ですので、半角で記述する文字列を全角で記述することはできません。

本文中に半角に変換されたくない文字列がある場合には、日本語記述のエスケープ文字で文字列を囲みます。詳細は“[8.3 日本語記述のエスケープ文字](#)”を参照してください。

変換規則

本文中の以下に示す文字は、無条件に半角に変換されます。

- 空白
- 特殊文字

本文中の空白、特殊文字およびシステム定義のキーワードで区切られる文字列が以下の文字だけで構成されている場合、半角に変換されます。

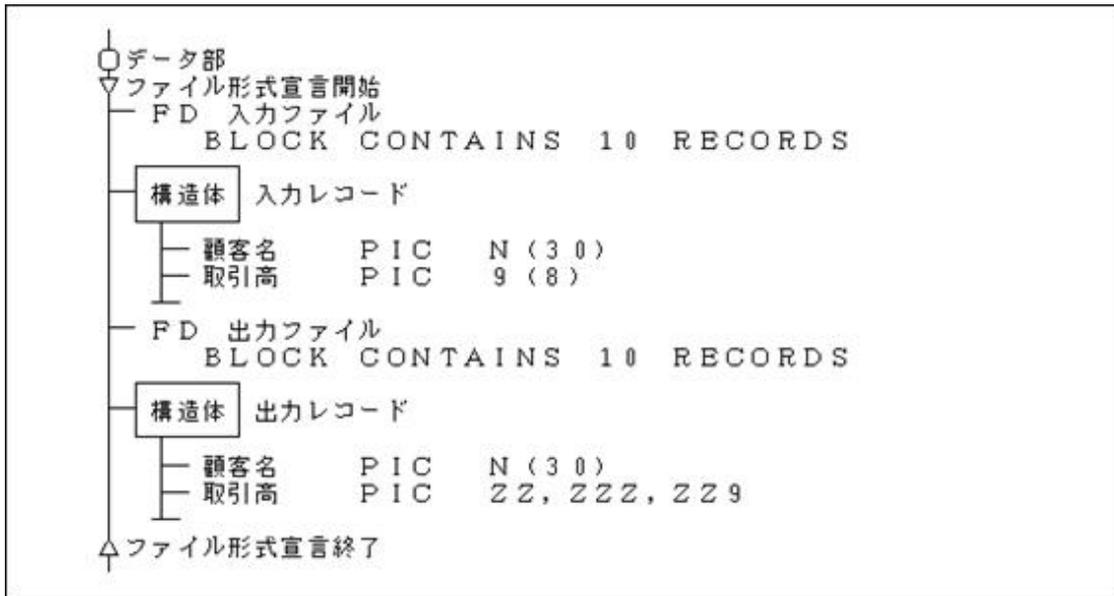
- 英大文字
- 英小文字
- 数字
- ハイフン(「-」)

特殊文字

ここでいう**特殊文字**とは、以下の文字を指しています。

+ (正符号)	* (アスタリスク)	/ (斜線)	= (等号)	¥ (円記号)	, (カンマ)
; (セミコロン)	. (ピリオド)	" (二重引用符)	((始め丸括弧)) (終わり丸括弧)	
> (不等号より大)	< (不等号より小)	: (コロン)	% (パーセント)	_ (アンダーライン)	
? (疑問符)	' (引用符)	@ (単位記号)	# (井げた)	\$ (ドル記号)	

記述例



第10章 段階的詳細化

本章では、段階的詳細化について説明します。

10.1 段階的詳細化とは

段階的詳細化とは、最初に処理の大枠(概要部)を作成し、続いて、大枠に基づいて詳細な処理(詳細部)を作成することをいいます。このようにプログラミングすることにより、自然にトップダウンプログラミングを実現することができます。

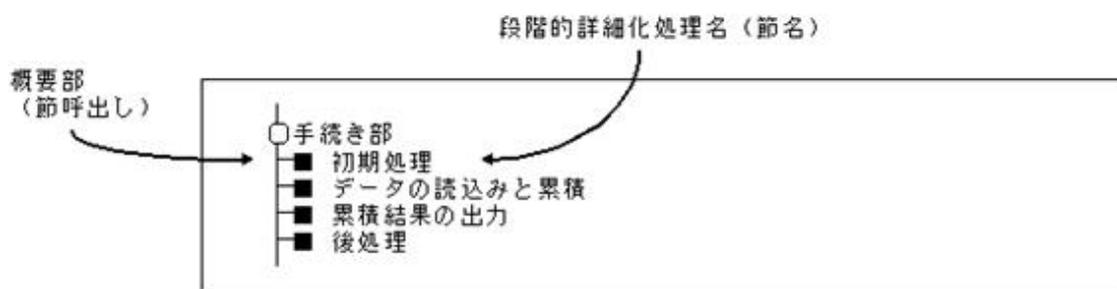
段階的詳細化を行うためには、節呼出しと節を使用します。先に説明した概要部に当たるのが節呼出しで、詳細部に当たるのが節です。また、節名のことを段階的詳細化処理名、節出口のことを段階的詳細化出口ともいいます。節呼出しおよび節の詳細については“3.10.6.7 節呼出し”および“3.9.2 節”を参照してください。

段階的詳細化によるプログラム例を“図10.1 段階的詳細化によるプログラム例”に示します。

図10.1 段階的詳細化によるプログラム例

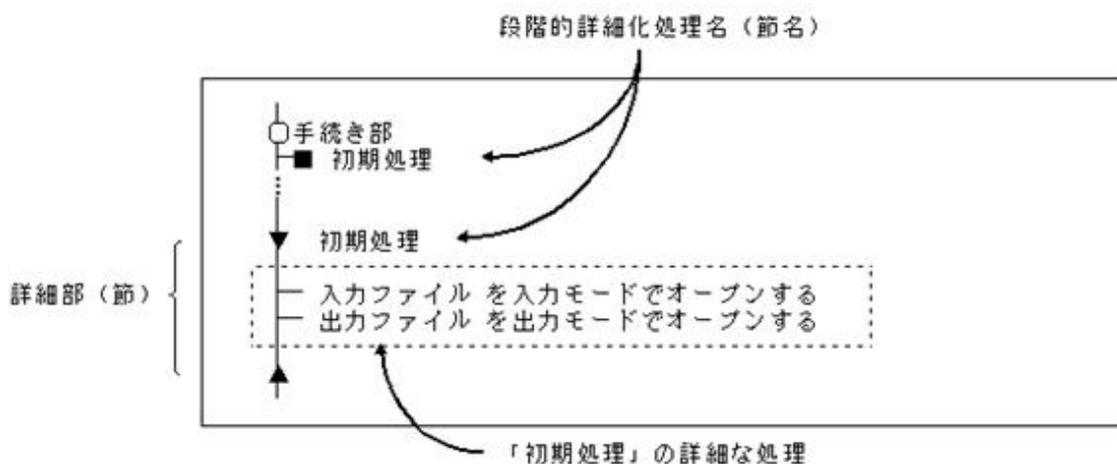
【ステップ 1 :】

プログラムの処理の大枠を概要部で記述します。



【ステップ 2 :】

各概要部で行う処理の詳細を詳細部で記述します。



付録A COBOL言語とYPS/COBOL言語の対応

ここでは、COBOL言語とYPS/COBOL言語の対応関係について説明しています。

A.1 翻訳単位

COBOL言語の翻訳単位は、YPS/COBOL言語のプログラム単位に対応します。対応関係を“表A.1 翻訳単位”に示します。

表A.1 翻訳単位

COBOL	YPS/COBOL
{[@OPTIONS COBOLオプションの並び] COBOL原始プログラム}	

A.2 COBOL原始プログラム

COBOL原始プログラムを構成する要素と対応するYPS表記を“表A.2 COBOL原始プログラムの構成”に示します。

表A.2 COBOL原始プログラムの構成

COBOL	記述の条件	YPS/COBOL
見出し部	COBOL原始プログラムが‘含まれるプログラム’でない場合	
	COBOL原始プログラムが‘含まれるプログラム’の場合	
[環境部]	—	[<input type="checkbox"/> 環境部]
[データ部]	—	[<input type="checkbox"/> データ部]
[手続き部]	—	<input type="checkbox"/> 手続き部
{[COBOL原始プログラム]}	—	
[プログラム終わり見出し]	—	

COBOL原始プログラムでは、手続き部およびプログラム終わり見出しを省略することができますが、YPS/COBOL言語では省略することができません。

A.3 見出し部とプログラム終わり見出し

COBOL言語の見出し部は、YPS/COBOL言語のプログラム開始または内部プログラム開始に対応します。

COBOL言語のプログラム終わり見出しは、YPS/COBOL言語のプログラム終了または内部プログラム終了に対応します。

見出し部

COBOL言語の見出し部とYPS/COBOL言語のプログラム開始または内部プログラム開始との対応関係を、“[表A.3 見出し部](#)”に示します。

表A.3 見出し部

COBOL	記述の条件	YPS/COBOL
IDENTIFICATION DIVISION. PROGRAM-ID. プログラム名 [IS 属性 PROGRAM].	COBOL原始プログラムが‘含まれるプログラム’でない場合	
	COBOL原始プログラムが‘含まれるプログラム’の場合	

見出し部の段落

次に示すCOBOL言語の段落を記述する場合には、プログラム開始または内部プログラム開始の直後にコメント文として記述してください。

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

USING指定

PROCEDURE DIVISIONのUSING指定で記述するパラメタは、プログラム開始または内部プログラム開始の本文に、パラメタとして記述します。パラメタには、COBOL言語のUSING指定の予約語USINGを除く文字列を記述します。

プログラム終わり見出し

COBOL言語のプログラム終わり見出しとYPS/COBOL言語のプログラム終了または内部プログラム終了との対応関係を、“[表A.4 プログラム終わり見出し](#)”に示します。

表A.4 プログラム終わり見出し

COBOL	YPS/COBOL
END PROGRAM プログラム名.	

A.4 環境部

COBOL言語の環境部は、YPS/COBOL言語の環境部に対応します。対応関係を“[表A.5 環境部](#)”に示します。

表A.5 環境部

COBOL	YPS/COBOL
ENVIRONMENT DIVISION.	□環境部

COBOL	YPS/COBOL
CONFIGURATION SECTION.	▽システム環境宣言開始 ⋮ △システム環境宣言終了
INPUT-OUTPUT SECTION.	▽入出力環境宣言開始 ⋮ △入出力環境宣言終了
SOURCE-COMPUTER.	├ SOURCE-COMPUTER
翻訳用計算機記述項	├ 翻訳用計算機記述項
OBJECT-COMPUTER.	├ OBJECT-COMPUTER
実行用計算機記述項	├ 実行用計算機記述項
SPECIAL-NAMES.	├ SPECIAL-NAMES
特殊名記述項	├ 特殊名記述項
SUBSCHEMA-NAME.	├ SUBSCHEMA-NAME
サブスキーマ名記述項	├ サブスキーマ名記述項
FILE-CONTROL.	├ FILE-CONTROL
ファイル管理記述項	├ ファイル管理記述項
I-O-CONTROL.	├ I-O-CONTROL
入出力管理記述項	├ 入出力管理記述項

COBOL言語の翻訳用計算機記述項、実行用計算機記述項、特殊名記述項、サブスキーマ名記述項、ファイル管理記述項および入出力管理記述項は、宣言文の本文にCOBOL言語の文法に従って記述します。ただし、記述項の終わりを示す終止符は、記述する必要はありません。宣言文については“3.8.1 宣言文”を参照してください。

段落の見出しと記述項を1つの宣言文の本文に記述することはできません。

A.5 データ部

COBOL言語のデータ部は、YPS/COBOL言語のプログラム内データ部に対応します。対応関係を“表A.6 データ部”に示します。

表A.6 データ部

COBOL	YPS/COBOL
DATA DIVISION.	□データ部
BASED-STORAGE SECTION.	▽基底変数宣言開始 ⋮ △基底変数宣言終了
FILE SECTION.	▽ファイル形式宣言開始 ⋮ △ファイル形式宣言終了
WORKING-STORAGE SECTION.	▽作業用データ宣言開始 ⋮ △作業用データ宣言終了
CONSTANT SECTION.	▽定数宣言開始 ⋮ △定数宣言終了

COBOL	YPS/COBOL
LINKAGE SECTION.	▽仮引数宣言開始 ⋮ △仮引数宣言終了
REPORT SECTION.	▽報告書式宣言開始 ⋮ △報告書式宣言終了
[Win] SCREEN SECTION.	▽画面宣言開始 ⋮ △画面宣言終了
ファイル記述項	├ ファイル記述項
整列併合用ファイル記述項	├ 整列併合用ファイル記述項
報告書記述項	├ 報告書記述項
独立データ記述項	├ 独立データ記述項
基本項目のデータ記述項 ・レコード記述項 ・報告集団記述項 ・[Win]画面データ記述項	├ [[データ名]] 任意なデータ句の並び ├ 66 データ名 RENAME句 ├ 88 条件名 VALUE句
集団項目のデータ記述項 ・レコード記述項 ・報告集団記述項 ・[Win]画面データ記述項	├ 構造体 [[データ名]] ├ [FILLER] ├ [<任意なデータ句の並び >]

データ部の段落の記述

宣言文の本文に記述するCOBOL言語の段落の見出しおよび記述項の終わりを表す終止符は、記述する必要はありません。

段落の見出しと記述項を1つの宣言文の本文に記述することはできません。

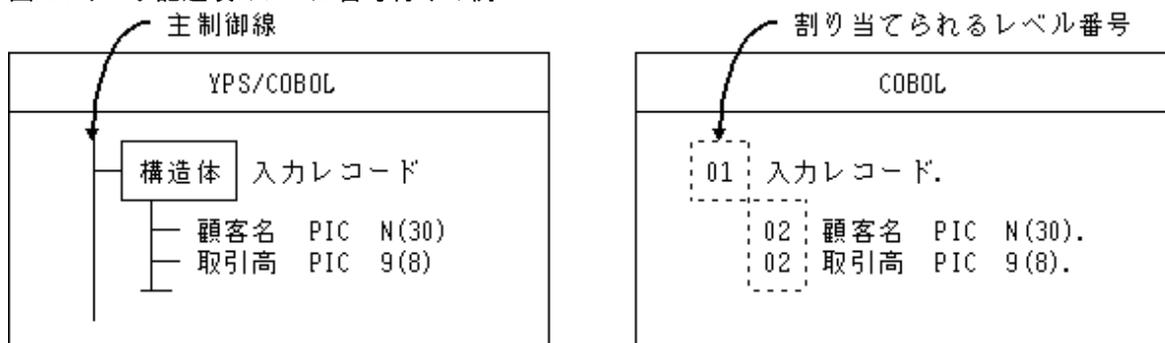
データ記述項のレベル番号付け

レベル番号66、77および88の項目を除くデータ記述項のレベル番号は記述する必要はありません。

YPS/COBOL言語の宣言文および構造体は、主制御線に位置する場合を01として、制御線のネストが深くなるにつれて1ずつ増加したレベル番号が割り当てられます。

データ記述項のレベル番号付けの例を“[図A.1 データ記述項のレベル番号付けの例](#)”に示します。

図A.1 データ記述項のレベル番号付けの例



A.6 手続き部

COBOL言語の手続き部は、YPS/COBOL言語の手続き部に対応します。対応関係を“表A.7 手続き部”に示します。

“表A.7 手続き部”でYPS/COBOL言語と対応関係がないCOBOL言語の文は、順次文または目的言語記述で記述してください。順次文および目的言語記述については、“3.10.1.1 順次文”および“3.12.4 目的言語記述”を参照してください。

また、“表A.7 手続き部”で示したYPS表記は、主制御線に記述した場合にはCOBOL言語の完結文に、主制御線以外の制御線に記述した場合にはCOBOL言語の完結文を構成する文に、それぞれ対応します。

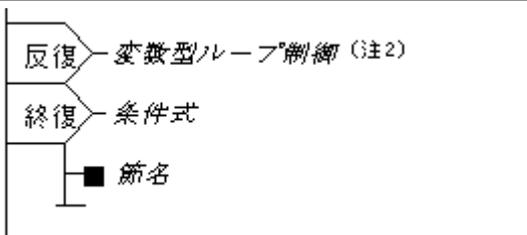
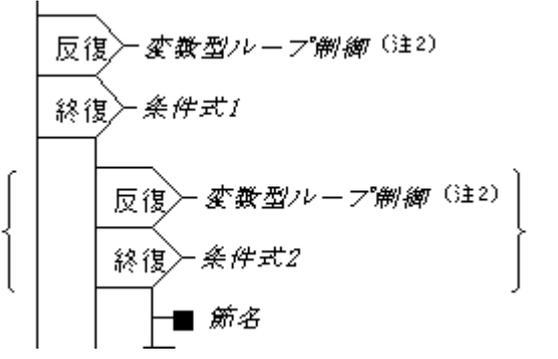
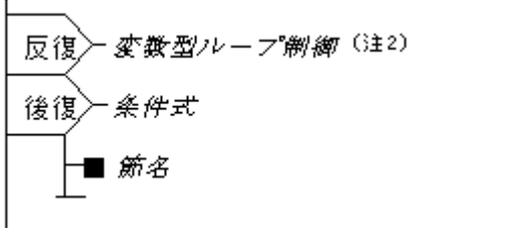
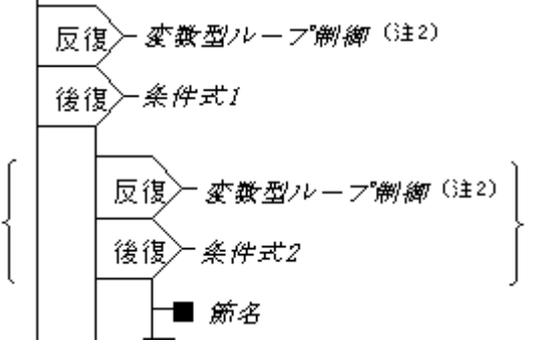
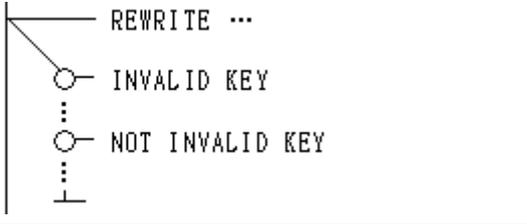
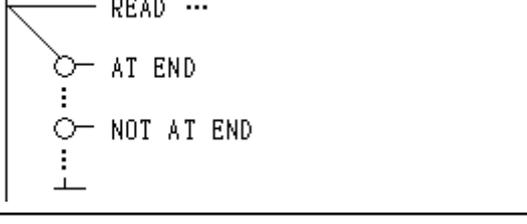
表A.7 手続き部

COBOL		YPS/COBOL
文	記述の条件	
PROCEDURE DIVISION [USING パラメタ] .	—	□ 手続き部
DECLARATIVES. ... END DECLARATIVES.	—	▽ 宣言部分開始 ⋮ △ 宣言部分終了
節の見出し	—	▼ 節名 ⋮ ▲
段落の見出し	—	▷ < 段落名 >
ADD文	ON SIZE ERROR指定および NOT ON SIZE ERROR指定を 行う場合	
CALL文	プログラム名が定数の場合	
	ON OVERFLOW指定を行う 場合	
	ON EXCEPTION指定および NOT ON EXCEPTION指定を 行う場合	

COBOL		YPS/COBOL
文	記述の条件	
COMPUTE文	ON SIZE ERROR指定および NOT ON SIZE ERROR指定を 行う場合	
CONTINUE文	—	└ <u>NONE</u>
DELETE文	INVALID KEY指定および NOT INVALID KEY指定を 行う場合	
DIVIDE文	ON SIZE ERROR指定および NOT ON SIZE ERROR指定を 行う場合	
EVALUATE文	—	
	選択主体の組が一意名1つで 選択対象が定数の場合	
ENTRY文	完結文の場合	
EXIT-PROGRAM文	—	

COBOL		YPS/COBOL
文	記述の条件	
FIND文 ^(註1)	AT END指定を行う場合	
GET文 ^(註1)	AT END指定を行う場合	
GO TO文	手続き名だけ指定する場合	
IF文 ^(註1)	THEN指定だけ行う場合	
	THEN指定およびELSE指定の両方を行う場合	
MULTIPLY文	ON SIZE ERROR指定およびNOT ON SIZE ERROR指定を行う場合	
うちPERFORM文	PERFORM (一意名 定数) TIMES 処理 END-PERFORM	
	PERFORM WITH TEST BEFORE UNTIL 条件式 処理 END-PERFORM	
	PERFORM WITH TEST BEFORE UNTIL NOT (条件式) 処理 END-PERFORM	

COBOL		YPS/COBOL
文	記述の条件	
	PERFORM WITH TEST AFTER UNTIL 条件式 処理 END-PERFORM	
	PERFORM WITH TEST BEFORE VARYING (一意名 指標名) FROM 初期値 BY 増分値 UNTIL 条件式 処理 END-PERFORM	
	PERFORM WITH TEST AFTER VARYING (一意名 指標名) FROM 初期値 BY 増分値 UNTIL 条件式 処理 END-PERFORM	
そとPERFORM文 ^(注3)	PERFORM 節名	
	PERFORM 節名 (一意名 指標名) TIMES	
	PERFORM 節名 WITH TEST BEFORE UNTIL 条件式	
	PERFORM 節名 THRU 節名 WITH TEST BEFORE UNTIL 条件式	
	PERFORM 節名 WITH TEST AFTER UNTIL 条件式	
	PERFORM 節名 THRU 節名 WITH TEST AFTER UNTIL 条件式	

COBOL		YPS/COBOL
文	記述の条件	
	PERFORM 節名 WITH TEST BEFORE VARYING (一意名 指標名) FROM 初期値 BY 増分値 UNTIL 条件式	
	PERFORM 節名 WITH TEST BEFORE VARYING (一意名 指標名 1) FROM 初期値1 BY 増分値1 UNTIL 条件式1 {AFTER (一意名2 指標名 2) FROM 初期値2 BY 増分値2 UNTIL 条件式2}	
	PERFORM 節名 WITH TEST AFTER VARYING (一意名 指標名) FROM 初期値 BY 増分値 UNTIL 条件式	
	PERFORM 節名 WITH TEST AFTER VARYING (一意名 指標名 1) FROM 初期値1 BY 増分値1 UNTIL 条件式1 {AFTER (一意名2 指標名 2) FROM 初期値2 BY 増分値2 UNTIL 条件式2}	
REWRITE文	INVALID KEY指定およびNOT INVALID KEY指定を行う場合	
READ文	AT END指定およびNOT AT END指定を行う場合	

COBOL		YPS/COBOL
文	記述の条件	
	INVALID KEY指定および NOT INVALID KEY指定を 行う場合	
SEARCH文 ^(注1)	—	
START文	INVALID KEY指定および NOT INVALID KEY指定を 行う場合	
STOP RUN文	—	
STORE文 ^(注1)	AT END指定を行う場合	
STRING文	ON OVERFLOW指定および NOT ON OVERFLOW指定を 行う場合	
SUBTRACT文	ON SIZE ERROR指定および NOT ON SIZE ERROR指定を 行う場合	
UNSTRING文	ON OVERFLOW指定および NOT ON OVERFLOW指定を 行う場合	

COBOL		YPS/COBOL
文	記述の条件	
WRITE文	INVALID KEY指定およびNOT INVALID KEY指定を行う場合	
	END-OF-PAGE指定およびNOT END-OF-PAGE指定を行う場合	

注1) YPS仕様書を記述場合に注意しなければならない点があります。“[A.7 YPS仕様書を記述する上での注意事項](#)”を参照してください。

注2) 変数型ループ制御については“[3.10.3.4 変数型ループ](#)”を参照してください。

注3) 手続き名として段落名を指定するそとPERFORM文は、節呼出しで記述することはできません。順次文を使用して、COBOL言語のPERFORM文で記述してください。

A.7 YPS仕様書を記述する上での注意事項

YPS仕様書を記述する場合には、以下の点に注意してください。

- COPY文およびREPLACE文で操作できないCOBOLの文
- COPY文およびREPLACE文の終止符の記述
- 埋込みSQLの記述
- 英小文字で記述することができないCOBOLの予約語
- YPSコンパイラが生成する段落の見出し
- AT END指定を行うFIND文、GET文またはSTORE文
- NEXT SENTENCE指定を行うIF文またはSEARCH文

COPY文およびREPLACE文で操作できないCOBOLの文

“[表A.8 COPY文およびREPLACE文で操作できないCOBOLの文](#)”に示すCOBOL言語の文は、COPY文により取り込んだり、REPLACE文により置き換えたりすることはできません。対応するYPS表記で記述してください。

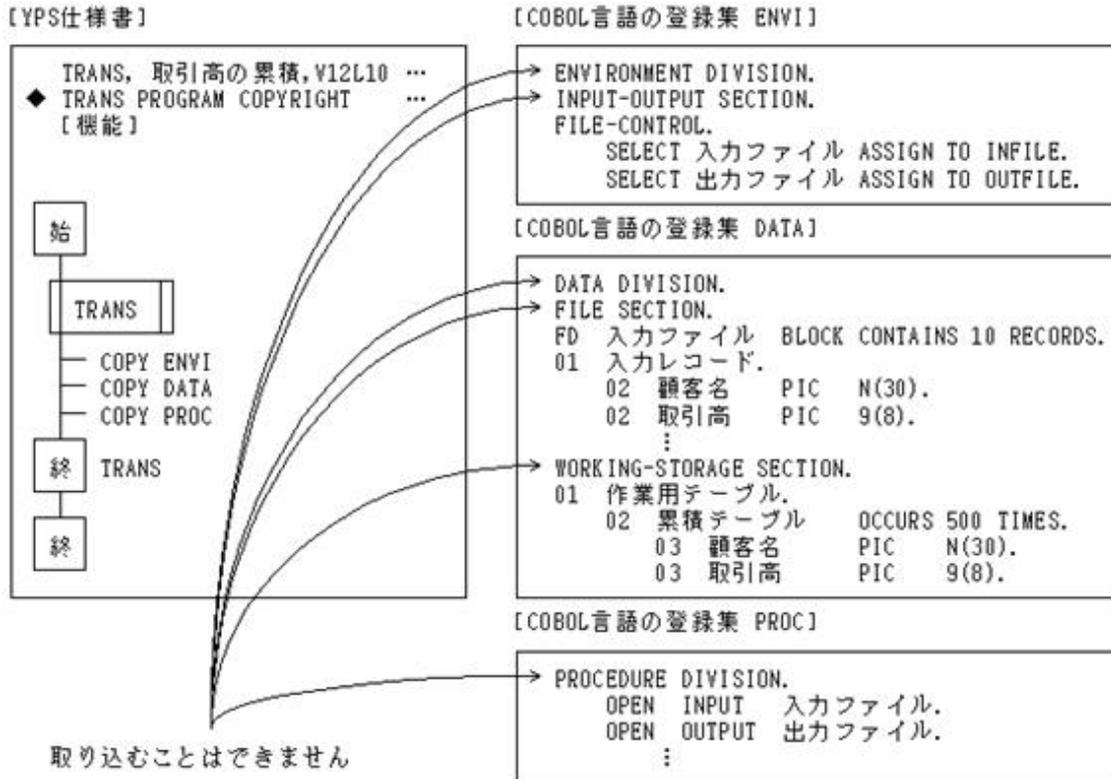
COPY文の誤った記述例を“[図A.2 COPY文の誤った記述例](#)”に、正しい記述例を“[図A.3 COPY文の正しい記述例](#)”に示します。

表A.8 COPY文およびREPLACE文で操作できないCOBOLの文

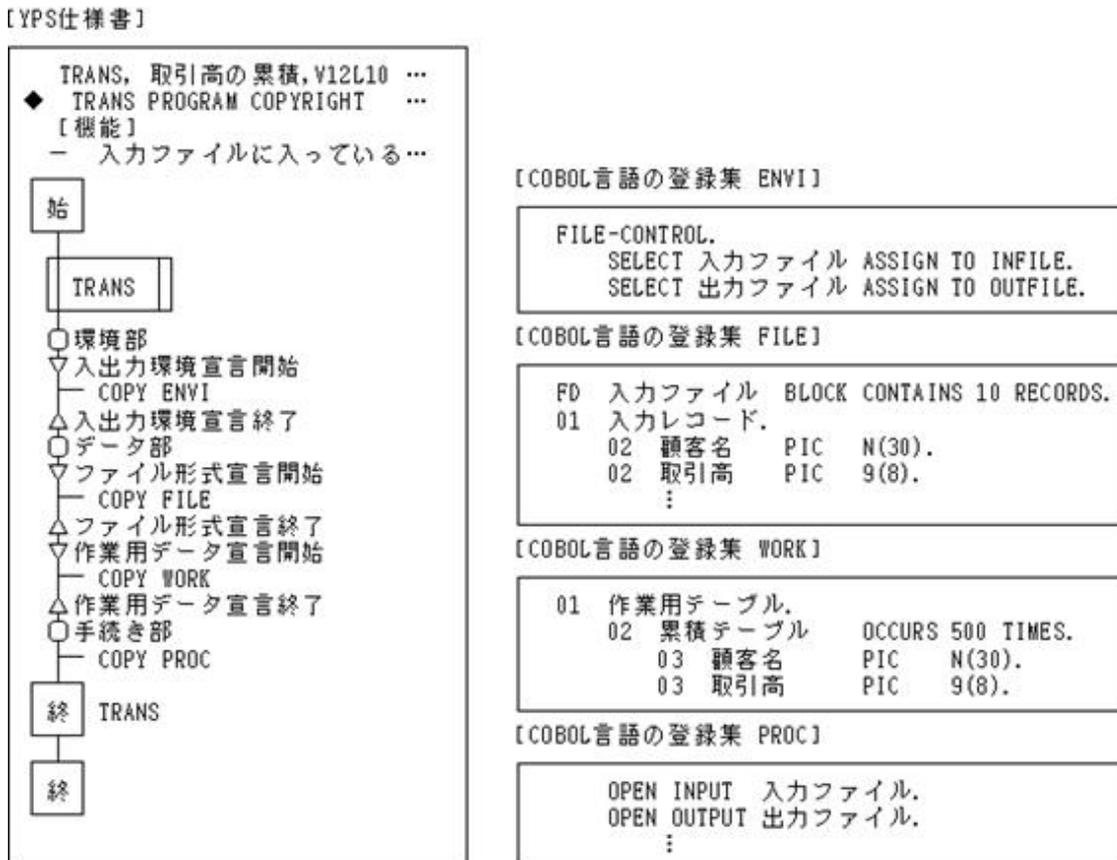
COBOLの文	YPS/COBOL言語の表記
IDENTIFICATION DIVISION. PROGRAM-ID. プログラム名.	
ENVIRONMENT DIVISION.	□環境部
CONFIGURATION SECTION.	▽システム環境宣言開始 ⋮ △システム環境宣言終了

COBOLの文	YPS/COBOL言語の表記
INPUT-OUTPUT SECTION.	▽入出力環境宣言開始 ⋮ △入出力環境宣言終了
DATA DIVISION.	□データ部
BASED-STORAGE SECTION.	▽基底変数宣言開始 ⋮ △基底変数宣言終了
FILE SECTION.	▽ファイル形式宣言開始 ⋮ △ファイル形式宣言終了
WORKING-STORAGE SECTION.	▽作業用データ宣言開始 ⋮ △作業用データ宣言終了
CONSTANT SECTION.	▽定数宣言開始 ⋮ △定数宣言終了
LINKAGE SECTION.	▽仮引数宣言開始 ⋮ △仮引数宣言終了
REPORT SECTION.	▽報告書式宣言開始 ⋮ △報告書式宣言終了
[Win] SCREEN SECTION.	▽画面宣言開始 ⋮ △画面宣言終了
PROCEDURE DIVISION.	□手続き部
DECLARATIVES. END DECLARATIVES.	▽宣言部分開始 ⋮ △宣言部分終了
END PROGRAM. プログラム名.	 プログラム名

図A.2 COPY文の誤った記述例



図A.3 COPY文の正しい記述例



COPY文およびREPLACE文の終止符の記述

目的言語記述以外の表記の本文にCOPY文およびREPLACE文を記述する場合にも、COPY文およびREPLACE文の終わりを表す終止符は利用者が記述してください。ただし、YPS/COBOLコンパイラが、宣言文または手続き部の主制御線上に記述した表記に付加する終止符で代用することもできます。

埋込みSQLの記述

埋込みSQLは、宣言文、順次文または目的言語記述で記述することができます。ただし、以下の点に注意してください。

- YPS仕様書中で使用できる引用符はアポストロフィ「'」です。ダブルクォーテーション「"」は使用できません。
- 埋込みSQLを抽象化処理構文定義の定義体内に記述する場合、一時変数を埋込みSQLのホスト変数として使用することはできません。ホスト変数はデータ部の埋込みSQL宣言節で宣言して使用してください。
- 宣言文または順次文で記述する場合、埋込みSQLの行内注記は記述できません。YPSのコメントの記述方法に従い記述してください。

埋込みSQLの記述例を“[図A.4 埋込みSQLの記述例](#)”に示します。

図A.4 埋込みSQLの記述例

```
├ EXEC SQL OPEN CUR1 END-EXEC
```

英小文字で記述することができないCOBOLの予約語

以下に示すCOBOLの予約語をYPS仕様書中に記述する場合には、英小文字または英大文字と英小文字の混在で記述することはできません。英大文字で記述してください。

- SOURCE-COMPUTER
- OBJECT-COMPUTER
- SPECIAL-NAMES
- SUBSCHEMA-NAME
- FILE-CONTROL
- I-O-CONTROL
- FD
- SD
- RD
- COPY
- REPLACE
- EXEC SQL

YPSコンパイラが生成する段落の見出し

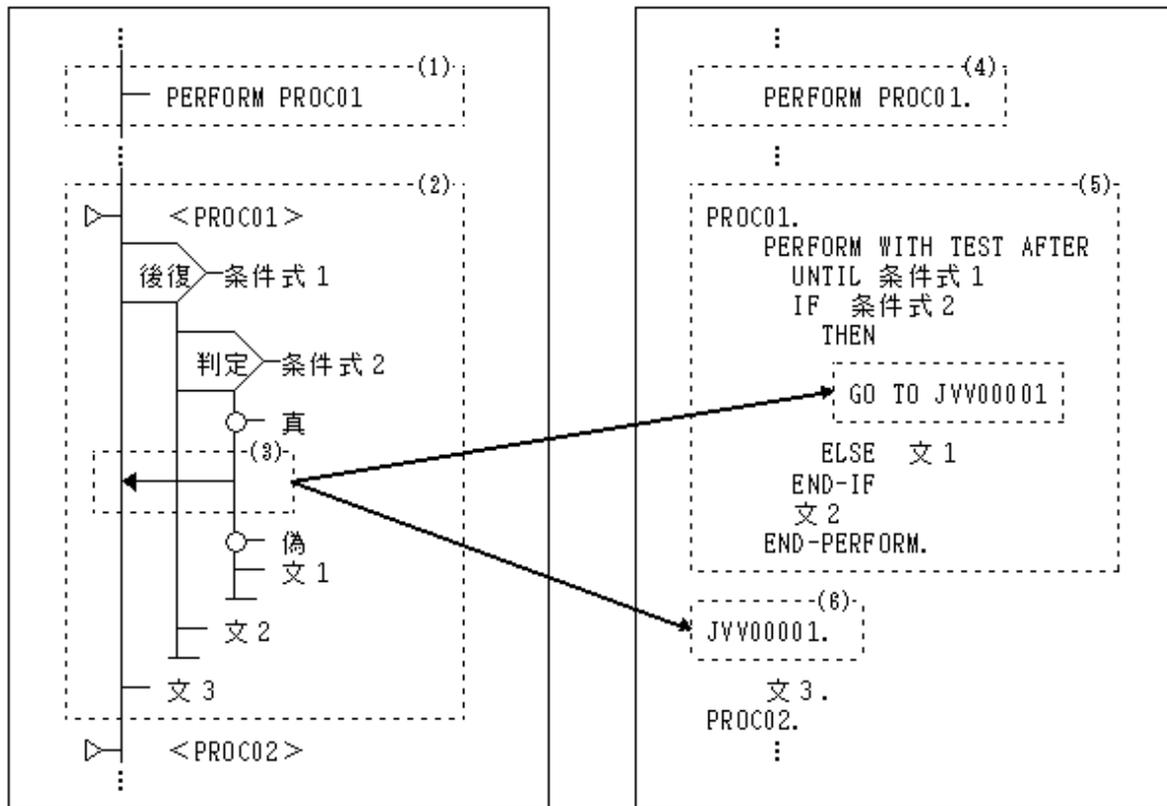
YPSコンパイラは、YPS仕様書からCOBOL原始プログラムを生成するときに、段落の見出しを生成する場合があります。よって、段落の見出しを生成する表記を含む段落(手続き)を、COBOL言語のPERFORM文、MERGE文およびSORT文で参照する場合には、YPSコンパイラが生成する段落の見出しのことを考慮する必要があります。

YPSコンパイラが生成した段落の見出しにより手続きの参照範囲が変わってしまう例を“[図A.5 YPSコンパイラが生成する段落の見出しにより手続きの参照範囲が変わる例](#)”に示します。

図A.5 YPSコンパイラが生成する段落の見出しにより手続きの参照範囲が変わる例

《YPS/COBOL言語の記述》

《生成されるCOBOL原始プログラム》



備考: (1)のPERFORM文で参照する手続きを(2)の範囲としても、YPSコンパイラの翻訳で(3)の途中脱出に対し、(6)の段落の見出しが生成されるので、(1)のPERFORM文の実行時に参照される手続きの範囲は(5)となり、文3は実行されません。

YPSコンパイラが段落の見出しを生成する表記と段落の見出しの生成位置を“表A.9 段落の見出しを生成する表記と段落の見出しの生成位置”に示します。

表A.9 段落の見出しを生成する表記と段落の見出しの生成位置

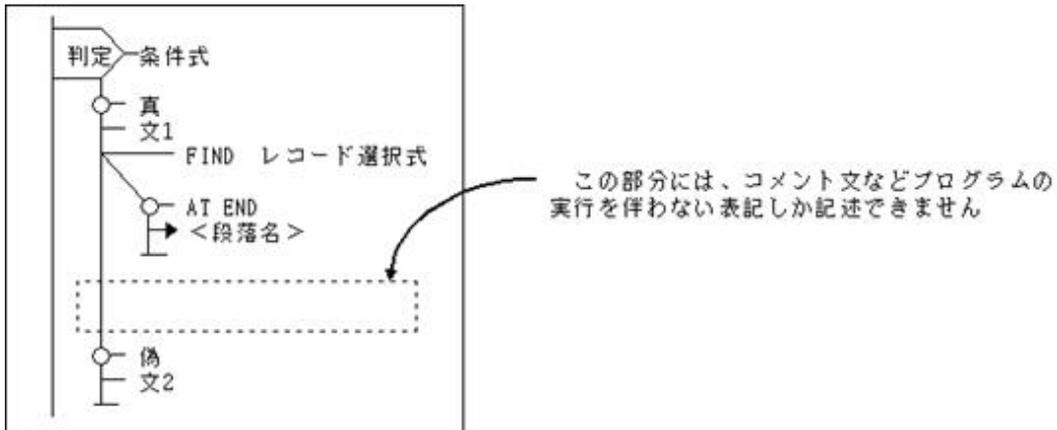
表記	段落の見出しの生成位置
更新型ループ	YPS表記を記述した場所の直前
無限ループ	
途中脱出	途中脱出を含む一番外側の選択系の表記またはループ系の表記の直後

AT END指定を行うFIND文、GET文またはSTORE文

FIND文、GET文またはSTORE文は、明示範囲符を持ちません。よって、これらの文を例外処理で記述し、判定の真節または偽節に記述する場合には、例外処理を真節または偽節の最後に記述しなければなりません。ただし、コメントを目的とした表記など、プログラムの実行を伴わない表記を例外処理の後に記述しても構いません。

“図A.6 [例外処理によるFIND文の記述]”に例外処理によるFIND文の記述例を示します。

図A.6 [例外処理によるFIND文の記述]



NEXT SENTENCE指定を行うIF文およびSEARCH文

NEXT SENTENCE指定を行うIF文およびSEARCH文は、判定または検索処理で記述することはできません。順次文または目的言語記述を利用してCOBOL言語のIF文またはSEARCH文で記述してください。

付録B 標準構文

標準構文は、利用者が定義しなくても利用できるようにシステムがあらかじめ用意した抽象化処理構文と抽象化条件式構文です。標準構文には、以下の三つがあります。

- YPS標準構文
- YPS/COBOL標準構文
- YPS/COBOL事務処理構文

B.1 YPS標準構文

YPS標準構文は、YPSの各言語に共通する標準構文です。

YPS標準構文には、抽象化処理構文しかありません。

YPS標準構文を“表B.1 YPS標準構文”に示します。

表B.1 YPS標準構文

抽象化処理構文	展開形(定義体)
【A】を行う	【A】
【A】に【B】を入れる	MOVE 【B】 TO 【A】
【A】を【B】に入れる	MOVE 【A】 TO 【B】
【A】に【B】を足す	COMPUTE 【A】 = 【A】 + 【B】
【A】を【B】に足す	COMPUTE 【B】 = 【B】 + 【A】
【A】から【B】を引く	COMPUTE 【A】 = 【A】 - 【B】
【A】を【B】から引く	COMPUTE 【B】 = 【B】 - 【A】
【A】を宣言する	【A】
【A】を取り込む	COPY 【A】 .

B.2 YPS/COBOL標準構文

YPS/COBOL標準構文は、YPS/COBOL言語の標準構文です。

YPS/COBOL標準構文には、抽象化処理構文と抽象化条件式構文があります。

YPS/COBOL標準構文の抽象化処理構文を“表B.2 YPS/COBOL標準構文の抽象化処理構文”に、YPS/COBOL標準構文の抽象化条件式構文を“表B.3 YPS/COBOL標準構文の抽象化条件式構文”に、それぞれ示します。

表B.2 YPS/COBOL標準構文の抽象化処理構文

種別	抽象化処理構文	展開形(定義体)
入力(ACCEPT文)	【A】を受け取る	ACCEPT 【A】
	【A】に年月日を格納する	ACCEPT 【A】 FROM DATE
	【A】に年日を格納する	ACCEPT 【A】 FROM DAY
	【A】に曜日を格納する	ACCEPT 【A】 FROM DAY-OF-WEEK
	【A】に時刻を格納する	ACCEPT 【A】 FROM TIME

種別	抽象化処理構文	展開形(定義体)
クローズ(CLOSE文)	【A】をクローズする	CLOSE 【A】
計算(COMPUTE文)	【B】 = 【A】	COMPUTE 【B】 = 【A】
削除(DELETE文)	【A】の現レコードを削除する	DELETE 【A】
表示(DISPLAY文)	【A】を表示する	DISPLAY 【A】
除算(DIVIDE文)	【A】を【B】で割る	DIVIDE 【B】 INTO 【A】
	【A】を【B】で割り商を【C】余りを【D】に入れる	DIVIDE 【B】 INTO 【A】 GIVING 【C】 REMAINDER 【D】
転記(MOVE文)	【A】 → 【B】	MOVE 【A】 TO 【B】
	【B】 ← 【A】	
乗算(MULTIPLY文)	【A】に【B】を掛ける	MULTIPLY 【B】 BY 【A】
オープン(OPEN文)	【A】を入力モードでオープンする	OPEN INPUT 【A】
	【A】を出力モードでオープンする	OPEN OUTPUT 【A】
	【A】を入出力モードでオープンする	OPEN I-O 【A】
	【A】を拡張モードでオープンする	OPEN EXTEND 【A】
読み込み(READ文)	【A】から次のレコードを読み込む	READ 【A】 NEXT
	【A】から【B】キー値のレコードを読み込む	READ 【A】 KEY IS 【B】
書換え(REWRITE文)	【A】で更新する	REWRITE 【A】
位置決め(START文)	【A】を【B】キー値に位置付ける	START 【A】 KEY IS EQUAL TO 【B】
	【A】を【B】より大きいキー値に位置付ける	START 【A】 KEY IS GREATER THAN 【B】
	【A】を【B】以上のキー値に位置付ける	START 【A】 KEY IS >= 【B】
停止(STOP文)	実行を終了する	STOP RUN
書き出し(WRITE文)	【A】を書き出す	WRITE 【A】
	【B】行送り後【A】を書き出す	WRITE 【A】 AFTER ADVANCING 【B】
	改ページ後【A】を書き出す	WRITE 【A】 AFTER ADVANCING PAGE

表B.3 YPS/COBOL標準構文の抽象化条件式構文

抽象化条件式構文	展開形(条件式定義体)
【A】と【B】が等しい	【A】 = 【B】
【A】と【B】が等しくない	【A】 NOT = 【B】
【A】が【B】より大きい	【A】 > 【B】
【A】が【B】より小さい	【A】 < 【B】
【A】が【B】以上	【A】 >= 【B】

抽象化条件式構文	展開形(条件式定義体)
【A】が【B】以下	【A】 <= 【B】
【A】が数字	【A】 IS NUMERIC
【A】が数字でない	【A】 IS NOT NUMERIC
【A】が英字	【A】 IS ALPHABETIC
【A】が英字でない	【A】 IS NOT ALPHABETIC
【A】がゼロ	【A】 IS ZERO
【A】がゼロでない	【A】 IS NOT ZERO

B.3 YPS/COBOL事務処理構文

YPS/COBOL事務処理構文は、事務処理に関する構文の集まりです。

YPS/COBOL事務処理構文には、抽象化処理構文と抽象化条件式構文があります。YPS/COBOL事務処理構文の抽象化処理構文は、標準日本語名標と組み合わせて使用します。標準日本語名標については、“付録C 標準日本語名標”を参照してください。

YPS/COBOL事務処理構文の抽象化処理構文を“表B.4 YPS/COBOL事務処理構文の抽象化処理構文”に、抽象化条件式構文を“表B.5 YPS/COBOL事務処理構文の抽象化条件式構文”に、それぞれ示します。

また、YPS/COBOL事務処理構文の抽象化処理構文と標準日本語名標を組み合わせた使用例を“表B.6 YPS/COBOL事務処理構文の抽象化処理構文の使用例”に示します。なお、使用例中の「」は、半角の空白列を表しています。

表B.4 YPS/COBOL事務処理構文の抽象化処理構文

種別	抽象化処理構文	展開形(定義体)
取消し (CANCEL文)	【A】をメモリから削除する	CANCEL 【A】
計算 (COMPUTE文)	【A】 = 【B】〔四捨五入〕	COMPUTE 【A】 ROUNDED = 【B】
除算(DIVIDE文)	【A】 ÷ 【B】 = 【C】 余り 【D】 とする	DIVIDE 【A】 BY 【B】 GIVING 【C】 REMAINDER 【D】
	【A】 ÷ 【B】 = 【C】 余り 【D】 〔四捨五入〕	DIVIDE 【A】 BY 【B】 GIVING 【C】 ROUNDED REMAINDER 【D】
初期化 (INITIALIZE文)	【A】を初期化する	INITIALIZE 【A】
	【A】を初期化す【る】 初期化条件は【B】とする	INITIALIZE 【A】 【B】
文字列検査 (INSPECT文)	【A】の文字列を数え【る】 計算項目は【B】とする	INSPECT 【A】 TALLYING 【B】
	【A】の文字列を置換す【る】 置換文字列は【B】とする	INSPECT 【A】 REPLACING 【B】
	【A】の文字列を数えて置換す【る】 計算項目は【B】とする	INSPECT 【A】 TALLYING 【B】
	【A】の文字を次の文字対応により変換す【る】 置換前文字列は【B】 置換後文字列は【C】とする	INSPECT 【A】 CONVERTING 【B】 TO 【C】

種別	抽象化処理構文	展開形(定義体)
併合 (MERGE文)	【A】と【B】を【昇降順】にマージす【る】 キー項目は【C】出力処理は【処理A】とする	MERGE 【A】 【昇降順】 【C】 USING 【B】 OUTPUT PROCEDURE 【処理A】
	【A】と【B】を【昇降順】にマージす【る】 キー項目は【C】出力ファイルは【D】とする	MERGE 【A】 【昇降順】 【C】 USING 【B】 GIVING 【D】
転記(MOVE 文)	【A】 → 【B】〔同一名転送〕	MOVE CORR 【A】 TO 【B】
	【A】をゼロクリアする	MOVE ZERO TO 【A】
	【A】をスペースクリアする	MOVE SPACE TO 【A】
オープン (OPEN文)	【A】を逆読みモードでオープンする	OPEN INPUT 【A】 REVERSED
読込み (READ文)	【A】を読み込む	READ 【A】 NEXT
	【A】のレコードを【B】に読み込む	READ 【A】 NEXT INTO 【B】
	【A】から【B】のレコードを読み込む	READ 【A】 KEY IS 【B】
引渡し (RELEASE 文)	ソートプログラムに【A】を引き渡す	RELEASE 【A】
	【A】を【B】に転送しソートプログラムに引き渡す	RELEASE 【B】 FROM 【A】
引取り (RETURN文)	【A】のソートマージレコードを受け取る	RETURN 【A】
	【A】のソートマージレコードを【B】に受け取る	RETURN 【A】 INTO 【B】
書換え (REWRITE 文)	【A】を更新する	REWRITE 【A】
	【A】を【B】に転送し更新する	REWRITE 【B】 FROM 【A】
	【A】を【B】に転送し【C】を更新する	REWRITE 【B】 OF 【C】 FROM 【A】
検索 (SEARCH文)	【A】を検索する	SEARCH 【A】
	【A】を【B】番目から検索する	SEARCH 【A】 VARYING 【B】
	【A】を二分探索法で検索する	SEARCH ALL 【A】
設定(SET文)	【A】を【B】にセットする	SET 【B】 TO 【A】
	【A】を【B】増やす	SET 【A】 UP BY 【B】
	【A】を【B】減らす	SET 【A】 DOWN BY 【B】
	【A】を指標とする	SET 【A】 TO TRUE
整列、分類 (SORT文)	【A】を【昇降順】にソートす【る】 キー項目は【B】 入力処理は【処理A】 出力処理は【処理B】とする	SORT 【A】 【昇降順】 【B】 WITH DUPLICATES INPUT PROCEDURE 【処理A】 OUTPUT PROCEDURE 【処理B】
	【A】を【昇降順】にソートす【る】 キー項目は【B】 入力ファイルは【C】 出力ファイルは【D】とする	SORT 【A】 【昇降順】 【B】 WITH DUPLICATES USING 【C】 GIVING 【D】

種別	抽象化処理構文	展開形(定義体)
位置決め (START文)	【A】を【B】のキー値に位置付ける	START 【A】 KEY IS EQUAL TO 【B】
連結 (STRING文)	文字列を結合す【る】 結合データは【A】 区切文字は【B】 受取項目は【C】とする	STRING 【A】 DELIMITED BY 【B】 INTO 【C】
	文字列を【A】文字目から結合す【る】 結合データは【B】 区切文字は【C】 受取項目は【D】とする	STRING 【B】 DELIMITED BY 【C】 INTO 【D】 WITH POINTER 【A】
分解 (UNSTRING文)	【A】の文字列を分解す【る】 区切文字は【B】 受取項目は【C】とする	UNSTRING 【A】 DELIMITED BY 【B】 INTO 【C】
	【A】の文字列を【B】文字目から分解す【る】 区切文字は【C】 受取項目は【D】とする	UNSTRING 【A】 DELIMITED BY 【C】 INTO 【D】 WITH POINTER 【B】
書き出し (WRITE文)	【A】を出力する	WRITE 【A】
	【A】を【B】に出力する	WRITE 【A】 OF 【B】
	【A】を【B】に転送して出力する	WRITE 【B】 FROM 【A】
	【A】を【B】に転送し【C】に出力する	WRITE 【B】 OF 【C】 FROM 【A】
	【A】を印刷後【B】行改行する	WRITE 【A】 BEFORE 【B】
	【A】を【B】に転送し印刷後【C】行改行する	WRITE 【B】 FROM 【A】 BEFORE 【C】
	【A】を【B】に転送し【C】に印刷後【D】行改行する	WRITE 【B】 OF 【C】 FROM 【A】 BEFORE 【D】
	【A】を印刷後改ページする	WRITE 【A】 BEFORE PAGE
	【A】を【B】に転送し【C】に印刷後改ページする	WRITE 【B】 OF 【C】 FROM 【A】 BEFORE PAGE
	【A】行改行後【B】を印刷する	WRITE 【B】 AFTER 【A】
	【A】行改行後【B】を【C】に転送して印刷する	WRITE 【C】 FROM 【B】 AFTER 【A】
	【A】行改行後【B】を【C】に転送し【D】に印刷する	WRITE 【C】 OF 【D】 FROM 【B】 AFTER 【A】
	改ページ後【A】を印刷する	WRITE 【A】 AFTER PAGE
	改ページ後【A】を【B】に転送して印刷する	WRITE 【B】 FROM 【A】 AFTER PAGE
改ページ後【A】を【B】に転送し【C】に印刷する	WRITE 【B】 OF 【C】 FROM 【A】 AFTER PAGE	

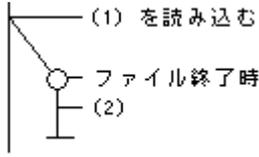
表B.5 YPS/COBOL事務処理構文の抽象化条件式構文

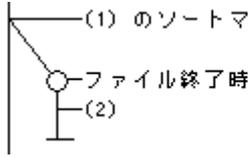
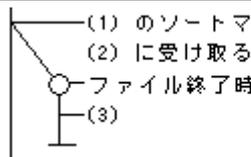
抽象化条件式構文	展開形(条件式定義体)
【変数1】 = 【変数2】	【変数1】 = 【変数2】

抽象化条件式構文	展開形(条件式定義体)
【変数1】 ≠ 【変数2】	【変数1】 NOT = 【変数2】
【変数1】 ≥ 【変数2】	【変数1】 >= 【変数2】
【変数1】 ≤ 【変数2】	【変数1】 <= 【変数2】
【変数1】 > 【変数2】	【変数1】 > 【変数2】
【変数1】 < 【変数2】	【変数1】 < 【変数2】
【A】 < 【変数】 ≤ 【C】	【A】 < 【変数】 AND 【変数】 <= 【C】
【A】 < 【変数】 < 【C】	【A】 < 【変数】 AND 【変数】 < 【C】
【A】 ≤ 【変数】 < 【C】	【A】 <= 【変数】 AND 【変数】 < 【C】
【A】 ≤ 【変数】 ≤ 【C】	【A】 <= 【変数】 AND 【変数】 = 【C】

表B.6 YPS/COBOL事務処理構文の抽象化処理構文の使用例

種別	使用例	展開形
取消し (CANCEL文)	(1) をメモリから削除する	DELETE (1)
計算 (COMPUTE文)	(1) = (2) ⊕ + ⊕ (3) ⊕ × ⊕ (4) ⊕ ÷ ⊕ (5) ⊕ …	COMPUTE (1) = (2) + (3) * (4) / (5) …
	(1) = (2) ⊕ - ⊕ (3) ⊕ ÷ ⊕ (4) ⊕ + ⊕ (5) ⊕ …	COMPUTE (1) = (2) - (3) / (4) + (5) …
	(1) = (2) ⊕ × ⊕ (3) ⊕ ÷ ⊕ (4) ⊕ … [四捨五入]	COMPUTE (1) ROUNDED = (2) * (3) / (4) …
	(1) = (2) ⊕ - ⊕ (3) ⊕ ÷ ⊕ (4) ⊕ … [四捨五入]	COMPUTE (1) ROUNDED = (2) - (3) / (4) …
除算(DIVIDE文)	(1) ÷ (2) = (3) 余り (4) とする	DIVIDE (1) BY (2) GIVING (3) REMAINDER (4)
	(1) ÷ (2) = (3) 余り (4) [四捨五入]	DIVIDE (1) BY (2) GIVING (3) ROUNDED REMAINDER (4)
初期化 (INITIALIZE文)	(1) を初期化する	INITIALIZE (1)
	(1) を初期化する 初期化条件は ⊕ 置き換え項目は ⊕ (2) ⊕ 転送データは ⊕ (3) ⊕ とする	INITIALIZE (1) REPLACING (2) DATA BY (3)
文字列検査 (INSPECT文)	(1) の文字数を数える ⊕ 計算項目は ⊕ (2) ⊕ 対象文字列は ⊕ $\left[\begin{array}{l} \text{全文字} \\ \left[\begin{array}{l} \text{全ての} \\ \left[\begin{array}{l} \text{ } \end{array} \right] \text{ (3)} \\ \text{先行する} \end{array} \right] \end{array} \right] \text{ (3)} \text{ ⊕$ [処理開始文字は ⊕ (4)] ⊕ [処理終了文字は ⊕ (5)] とする	INSPECT (1) TALLYING (2) FOR $\left[\begin{array}{l} \text{CHARACTERS} \\ \left[\begin{array}{l} \text{ALL} \\ \left[\begin{array}{l} \text{ } \end{array} \right] \end{array} \right] \end{array} \right] \text{ (3)}$ [AFTER (4)] [BEFORE (5)]

種別	使用例	展開形
	(1) の文字列を置換する□ 置換文字列は□ $\left[\begin{array}{l} \text{全文字} \\ \left[\begin{array}{l} \text{全ての} \\ \text{先行する} \end{array} \right] \square(2) \end{array} \right] \square$ 置換後文字列は□(3)□ [処理開始文字は□(4)]□ [処理終了文字は□(5)]とする	INSPECT (1) REPLACING CHARACTERS $\left[\begin{array}{l} \text{ALL} \\ \left[\begin{array}{l} \text{LEADING} \\ \text{FIRST} \end{array} \right] (2) \end{array} \right]$ BY (3) [AFTER (4)] [BEFORE (5)]
	(1) の文字列を数えて置換する 計算項目は□(2)□ 対象文字列は□ $\left[\begin{array}{l} \text{全文字列} \\ \left[\begin{array}{l} \text{全ての} \\ \text{先行する} \end{array} \right] \square(3) \end{array} \right] \square$ [処理開始文字は□(4)]□ [処理終了文字は□(5)]□ 置換前文字列は□ $\left[\begin{array}{l} \text{全文字列} \\ \left[\begin{array}{l} \text{全ての} \\ \text{先行する} \end{array} \right] \square(6) \end{array} \right] \square$ 置換後文字列は□(7)□ [処理開始文字は□(8)]□ [処理終了文字は□(9)]とする	INSPECT (1) TALLYING (2) FOR $\left[\begin{array}{l} \text{CHARACTERS} \\ \left[\begin{array}{l} \text{ALL} \\ \left[\begin{array}{l} \text{LEADING} \end{array} \right] (3) \end{array} \right] \end{array} \right]$ [AFTER (4)] [BEFORE (5)] REPLACING $\left[\begin{array}{l} \text{CHARCTERS} \\ \left[\begin{array}{l} \text{ALL} \\ \left[\begin{array}{l} \text{LEADING} \end{array} \right] (6) \\ \text{FIRST} \end{array} \right] \end{array} \right]$ BY (7) [AFTER (8)] [BEFORE (9)]
	(1) の文字列を次の文字対応により変換する□ 置換前文字列は□(2)□ 置換後文字列は□(3)□ [処理開始文字列は□(4)]□ [処理終了文字列は□(5)]とする	INSPECT (1) CONVERTING (2) TO (3) [AFTER (4)] [BEFORE (5)]
併合 (MERGE文)	(1) と (2) を $\left[\begin{array}{l} \text{昇順} \\ \text{降順} \end{array} \right]$ にマージする□ キー項目は□(3)□(4)□(5)□…□ 出力処理は□(6)とする	MERGE (1) $\left[\begin{array}{l} \text{ASCENDING KEY} \\ \text{DESCENDING KEY} \end{array} \right] (3) (4) (5) \dots$ USING (2) OUTPUT PROCEDURE (6)
	(1) と (2) を $\left[\begin{array}{l} \text{昇順} \\ \text{降順} \end{array} \right]$ にマージする□ キー項目は□(3)□(4)□(5)□…□ 出力ファイルは□(6)とする	MERGE (1) $\left[\begin{array}{l} \text{ASCENDING KEY} \\ \text{DESCENDING KEY} \end{array} \right] (3) (4) (5) \dots$ USING (2) GIVING (6)
転記(MOVE文)	(1) → (2) [同一名転送]	MOVE CORR (1) TO (2)
	(1) をゼロクリアする	MOVE ZERO TO (1)
	(1) をスペースクリアする	MOVE SPACE TO (1)
読み込み (READ文)	 (1) を読み込む ファイル終了時 (2)	READ (1) NEXT AT END (2)
	(1) のレコードを (2) に読み込む	READ (1) NEXT INTO (2)

種別	使用例	展開形
	(1) から (2) のレコードを読み込む	READ (1) KEY IS (2)
引渡し (RELEASE 文)	ソートプログラムに (1) を引き渡す	RELEASE (1)
	(1) を (2) に転送しソートプログラムに引き渡す	RELEASE (2) FROM (1)
引取り (RETURN文)	 (1) のソートマージレコードを受け取る ファイル終了時 (2)	RETURN (1) AT END (2)
	 (1) のソートマージレコードを (2) に受け取る ファイル終了時 (3)	RETURN (1) INTO (2) AT END (3)
書換え (REWRITE 文)	(1) を更新する	REWRITE (1)
	(1) を (2) に転送し更新する	REWRITE (2) FROM (1)
	(1) を (2) に転送し (3) を更新する	REWRITE (2) OF (3) FROM (1)
検索 (SEARCH文)	 (1) を検索する 検索終了時 (2) 検索条件 (3) (4) 検索条件 (5) (6)	SEARCH (1) AT END (2) WHEN (3) (4) WHEN (5) (6) END-SEARCH
	(1) を (2) 番目から検索する	SEARCH (1) VARYING (2)
	(1) を二分探索法で検索する	SEARCH ALL (1)
設定(SET文)	(1) を (2) にセットする	SET (2) INTO (1)
	(1) を (2) 増やす	SET (1) UP BY (2)
	(1) を (2) 減らす	SET (1) DOWN BY (2)
	(1) を指標とする	SET (1) TO TRUE
整列、分類 (SORT文)	(1) を $\left[\begin{array}{l} \text{昇順} \\ \text{降順} \end{array} \right]$ にソートする キー項目は (2) (3) (4) … 入力処理は (4) 出力処理は (5) とする	SORT (1) $\left[\begin{array}{l} \text{ASCENDING KEY} \\ \text{DESCENDING KEY} \end{array} \right]$ (2) (3) … WITH DUPLICATES INPUT PROCEDURE (4) OUTPUT PROCEDURE (5)
	(1) を $\left[\begin{array}{l} \text{昇順} \\ \text{降順} \end{array} \right]$ にソートする キー項目は (2) (3) (4) … 入力ファイルは (4) 出力ファイルは (5) とする	SORT (1) $\left[\begin{array}{l} \text{ASCENDING KEY} \\ \text{DESCENDING KEY} \end{array} \right]$ (2) (3) … WITH DUPLICATES USING (4) GIVING (5)

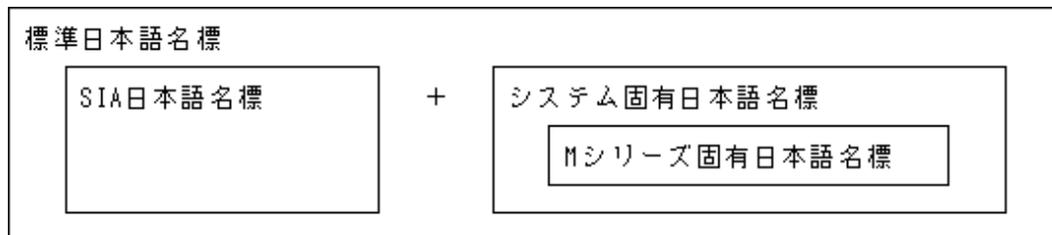
種別	使用例	展開形
位置決め (START文)	(1) を (2) のキー値に位置付ける	START (1) KEY IS EQUAL TO (2)
連結 (STRING文)	文字列を結合する□ 結合データは□(1)□(2)□(3)□…□ 区切文字は□(4)□受取項目は□(5)とする	STRING (1) (2) (3) ... DELIMITED BY (4) INTO (5)
	文字列を (1) 文字目から結合する□ 結合データは□(2)□(3)□(4)□…□ 区切文字は□(5)□受取項目は□(6)とする	STRING (2) (3) (4) ... DELIMITED BY (5) INTO (6) WITH POINTER (1)
分解 (UNSTRING文)	(1) の文字列を分解する□ 区切文字は□(2)□OR□(3)□OR□(4)□…□ 受取項目は□(5)□(6)□(7)□…とする	UNSTRING (1) DELIMITED BY (2) OR (3) OR (4) ... INTO (5) (6) (7) ...
	(1) の文字列を分解する□ 区切文字は□(2)□OR□(3)□OR□(4)□…□ 受取項目は□(5)□ カウント項目は□(6)□ 区切文字項目は□(7)□ ... (8)□ カウント項目は□(9)□ 区切文字項目は□(10)□ ... 計算項目は□(11)とする	UNSTRING (1) DELIMITED BY (2) OR (3) OR (4) ... INTO (5) COUNT IN (6) DELIMITER (7) ... (8) COUNT IN (9) DELIMITER IN (10) TALLYING (11)
	(1) の文字列を (2) 文字目から分解する□ 区切文字は□(3)□OR□(4)□OR□(5)□…□ 受取項目は□(6)とする	UNSTRING (1) DELIMITED BY (3) OR (4) OR (5) ... INTO (6) WITH POINTER (2)
書き出し (WRITE文)	(1) を出力する	WRITE (1)
	(1) を (2) に出力する	WRITE (1) OF (2)
	(1) を (2) に転送して出力する	WRITE (2) FROM (1)
	(1) を (2) に転送し (3) に出力する	WRITE (2) OF (3) FROM (1)
	(1) を印刷後 (2) 行改行する	WRITE (1) BEFORE (2)
	(1) を (2) に転送し印刷後 (3) 行改行する	WRITE (2) FROM (1) BEFORE (3)
	(1) を (2) に転送し (3) に印刷後 (4) 行改行する	WRITE (2) OF (3) FROM (1) BEFORE (4)
	(1) を印刷後改ページする	WRITE (1) BEFORE PAGE
	(1) を (2) に転送し (3) に印刷後改ページする	WRITE (2) OF (3) FROM (1) BEFORE PAGE
	(1) 行改行後 (2) を印刷する	WRITE (2) AFTER (1)
	(1) 行改行後 (2) を (3) に転送して印刷する	WRITE (3) FROM (2) AFTER (1)
	(1) 行改行後 (2) を (3) に転送し (4) に印刷する	WRITE (3) OF (4) FROM (2) AFTER (1)
	改ページ後 (1) を印刷する	WRITE (1) AFTER PAGE
	改ページ後 (1) を (2) に転送して印刷する	WRITE (2) FROM (1) AFTER PAGE
改ページ後 (1) を (2) に転送し (3) に印刷する	WRITE (2) OF (3) FROM (1) AFTER PAGE	

付録C 標準日本語名標

標準日本語名標は、利用者が定義しなくても利用できるようにシステムがあらかじめ用意した日本語名標です。

標準日本語名標には、“[図C.1 標準日本語名標の構成](#)”に示すように、各システムで共通して使用できるSIA日本語名標と、各システムに固有なシステム固有日本語名標とがあります。システム固有日本語名標は、Mシリーズ固有日本語名標だけを提供しています。

図C.1 標準日本語名標の構成



C.1 SIA日本語名標

SIA日本語名標は、各システムで共通して使用できる標準日本語名標です。

SIA日本語名標を“[表C.1 SIA日本語名標](#)”に示します。

表C.1 SIA日本語名標

主に使用する場所	日本語名標(文字列1)	COBOLの文字列(文字列2)
システム環境宣言節	文字の大小順序は	PROGRAM COLLATING SEQUENCE IS
	特殊名段落	SPECIAL-NAMES
入出力環境宣言節	入出力ファイル指定段落	FILE-CONTROL
	ファイル名は	;
	ファイル指定	SELECT
	ファイル識別名は	ASSIGN TO
	ファイル種別は	ORGANIZATION IS
	索引ファイル	INDEXED
	相対ファイル	RELATIVE
	順ファイル	SEQUENTIAL
	呼出しモードは	ACCESS MODE
	順呼出し	SEQUENTIAL
	乱呼出し	RANDOM
	動的呼出し	DYNAMIC
	レコードキーは	RECORD KEY IS
	主キー項目は	RECORD KEY IS
相対キー項目は	RELATIVE KEY IS	
重複キーあり	WITH DUPLICATES	

主に使用する場所	日本語名標(文字列1)	COBOLの文字列(文字列2)
	バッファ数は	RESERVE
	ファイル状態変数は	FILE STATUS IS
	入出力管理段落	I-O-CONTROL
	共用ファイル領域は	SAME AREA
	共用レコード領域は	SAME RECORD AREA
ファイル形式宣言節	ファイル形式定義	FD
	最小レコード長は	FROM
	最大レコード長は	TO
	可変長	VARYING
	レコード長項目は	DEPENDING ON
	ブロック数は	BLOCK CONTAINS
	ブロック長は	BLOCK
	レコード長は	RECORD
	レコード	RECORDS
	バイト	CHARACTERS
	ソートファイル形式定義	SD
	マージファイル形式定義	SD
	外部属性指定	IS EXTERNAL
	全域指定	IS GLOBAL
データ部	-----	FILLER
	データ形式は	PIC
	初期値は	VALUE
	再定義名は	REDEFINES
	反復回数は	OCCURS
	指標名は	INDEXED BY
	表制御変数は	DEPENDING ON
	昇順キーは	ASCENDING KEY
	降順キーは	DESCENDING KEY
	数字モードは	USAGE IS
	外部10進モード	DISPLAY
	内部10進モード	PACKED-DECIMAL

主に使用する場所	日本語名標(文字列1)	COBOLの文字列(文字列2)
	2進モード	COMPUTATIONAL
	転送データは	DATA BY
	英字項目	ALPHABETIC
	英数字項目	ALPHANUMERIC
	数字項目	NUMERIC
	英数字編集項目	ALPHANUMERIC-EDITED
	数字編集項目	NUMERIC-EDITED
	日本語項目	NATIONAL
	日本語編集項目	NATIONAL-EDITED
	指標モード	INDEX
	境界調整指定	SYNCHRONIZED
	符号位置は	SIGN IS
	先頭桁	LEADING
	最終桁	TRAILING
	分離モード指定	SEPARATE CHARACTER
	右詰め指定	JUSTIFIED RIGHT
	ゼロを空白にする	BLANK WHEN ZERO
	12ポ全角	MODE-1
	9ポ全角	MODE-2
	7ポ全角	MODE-3
	空白	SPACE
	ゼロ	ZERO
手続き部	昇順	ON ASCENDING KEY
	降順	ON DESCENDING KEY
	完了コード	PROGRAM-STATUS
	桁あふれ時	ON SIZE ERROR
	桁あふれなし時	NOT ON SIZE ERROR
	ファイル終了時	AT END
	ファイル終了時以外	NOT AT END
	ページ終了時	AT EOP
	無効キー時	INVALID KEY

主に使用する場所	日本語名標(文字列1)	COBOLの文字列(文字列2)
	正常キー時	NOT INVALID KEY
	検索条件	WHEN
	検索終了時	AT END
	検索終了	END-SEARCH
	該当なし	AT END
		ALSO
	～	THRU
	--	ANY
	ソート完了コード	SORT-STATUS
	マージ完了コード	SORT-STATUS
	処理続行	CONTINUE
	次処理へ	NEXT SENTENCE
	エラー手続き宣言部	DECLARATIVES
	宣言終了	END DECLARATIVES
	入力モード用	INPUT
	出力モード用	OUTPUT
	入出力モード用	I-O
	拡張モード用	EXTEND
STRING文	あふれ時	ON OVERFLOW
	あふれなし	NOT ON OVERFLOW
	結合終了	END-STRING
	分解終了	END-UNSTRING
UNSTRING文	区切文字項目は	DELIMITER IN
	カウント項目は	COUNT IN
	計算項目は	TALLYING
INSPECT文	最初の	FIRST
	先行する	LEADING
	全ての	ALL
	対象文字列は	FOR
	全文字	CHARACTERS
	処理開始文字は	AFTER

主に使用する場所	日本語名標(文字列1)	COBOLの文字列(文字列2)
	処理終了文字は	BEFORE
	置換後文字列は	BY
	置換前文字列は	REPLACING
INITIALIZE文	置き換え項目は	REPLACING
COMPUTE文	÷	/
	×	*
	+	+
	-	-

C.2 Mシリーズ固有日本語名標 [GS]

Mシリーズ固有日本語名標は、GS/Mシリーズだけで利用できる標準日本語名標です。

Mシリーズ固有日本語名標を“表C.2 Mシリーズ固有日本語名標”に示します。

表C.2 Mシリーズ固有日本語名標

主に使用する場所	日本語名標(文字列1)	COBOLの文字列(文字列2)
データ部	半角	HSC
	倍角	F0202
	半角の倍角	H0202
	長体	F0102
	平体	F0201
	副キー項目は	ALTERNATE RECORD KEY IS
	符号系は	CODE-SET
	図表レコードは	FORMATTED RECORD
	ビットモード	BIT
	ブール項目	BOOLEAN
	1 2 ポ半角	MODE-1 BY HSC
	9 ポ半角	MODE-2 BY HSC
	1 2 ポ倍角	MODE-1 BY F0202
	9 ポ倍角	MODE-2 BY F0202
	7 ポ倍角	MODE-3 BY F0202
	1 2 ポ半角の倍角	MODE-1 BY H0202
	9 ポ半角の倍角	MODE-2 BY H0202
	1 2 ポ長体	MODE-1 BY F0102

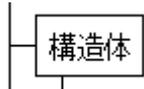
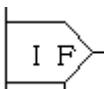
主に使用する場所	日本語名標(文字列1)	COBOLの文字列(文字列2)
	9 ポ長体	MODE-2 BY F0102
	7 ポ長体	MODE-3 BY F0102
	1 2 ポ平体	MODE-1 BY F0201
	9 ポ平体	MODE-2 BY F0201
	7 ポ平体	MODE-3 BY F0201

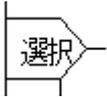
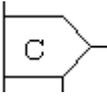
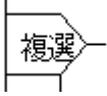
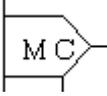
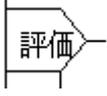
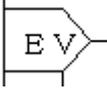
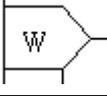
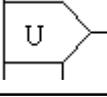
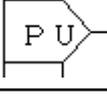
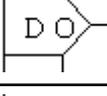
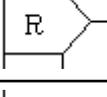
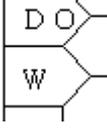
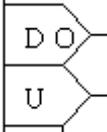
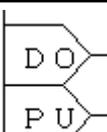
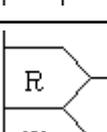
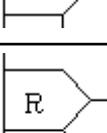
付録D 日本語図記号と英語図記号の対応表

YPS/COBOL言語で用意されている図記号には、日本語と英語の2種類があります。日本語と英語で異なる図記号について“表D.1 日本語図記号と英語図記号の対応表”に示します。

表D.1 日本語図記号と英語図記号の対応表

図記号名	日本語図記号	英語図記号
モジュール本体開始	始	S
終了	終	E
翻訳指示行	◇ 翻訳 :	◇ CI:
構文定義部開始	□ 構文定義部	□ SYNTAX DEFINITION
環境部開始	□ 環境部	□ ENVIRONMENT DIVISION
データ部開始	□ データ部	□ DATA DIVISION
手続き部開始	□ 手続き部	□ PROCEDURE DIVISION
宣言部分開始	▽ 宣言部分開始	▽ DECLARATIVES
宣言部分終了	△ 宣言部分終了	△ END DECLARATIVES
システム環境宣言開始	▽ システム環境宣言開始	▽ CONFIGURATION
システム環境宣言終了	△ システム環境宣言終了	△ END CONFIGURATION
入出力環境宣言開始	▽ 入出力環境宣言開始	▽ INPUT-OUTPUT
入出力環境宣言終了	△ 入出力環境宣言終了	△ END INPUT-OUTPUT
基底変数宣言開始	▽ 基底変数宣言開始	▽ BASED-STORAGE

図記号名	日本語図記号	英語図記号
基底変数宣言終了	 基底変数宣言終了	 END BASED-STORAGE
ファイル形式宣言開始	 ファイル形式宣言開始	 FILE
ファイル形式宣言終了	 ファイル形式宣言終了	 END FILE
作業用データ宣言開始	 作業用データ宣言開始	 WORKING-STORAGE
作業用データ宣言終了	 作業用データ宣言終了	 END WORKING-STORAGE
定数宣言開始	 定数宣言開始	 CONSTANT
定数宣言終了	 定数宣言終了	 END CONSTANT
仮引数宣言開始	 仮引数宣言開始	 PARAMETER
仮引数宣言終了	 仮引数宣言終了	 END PARAMETER
報告書式宣言開始	 報告書式宣言開始	 REPORT
報告書式宣言終了	 報告書式宣言終了	 END REPORT
[Win] 画面宣言開始	 画面宣言開始	 SCREEN
[Win] 画面宣言終了	 画面宣言終了	 END SCREEN
構造体		
判定		

図記号名	日本語図記号	英語図記号
選択		
複合選択		
評価		
前復		
後復		
終復		
反復		
更復		
反復と前復の組合せ		
反復と後復の組合せ		
反復と終復の組合せ		
更復と前復の組合せ		
更復と後復の組合せ		

図記号名	日本語図記号	英語図記号
注釈開始	★注釈開始	★COMMENT
注釈終了	★注釈終了	★END COMMENT
要約		
目的言語		
条件付き翻訳判定		
条件付き翻訳選択		
取込み		
抽象化処理構文定義開始		

付録E YPS仕様書のファイル形式とコード系

システムごとのYPS仕様書のファイル形式とコード系を“表E.1 システムごとのYPS仕様書のファイル形式とコード系”に示します。

表E.1 システムごとのYPS仕様書のファイル形式とコード系

システム名		ファイル形式	最大レコード長	最大文字数/行 (バイト)	コード系	
					日本語	その他
G S	F4 MSP	VB	255	120	JEF	EBCDIC(カナ)/
	MSP					EBCDIC(英小文字)
	XSP					
Windows		-	252		シフトJIS	ASCII

- ・GSとその他で最大レコード長が異なるのは、可変長レコードの形式が異なるためです。

ー GS:

RDW(4バイト) + レコードデータ

ー ASP、CSP/FX:

RDI(2バイト) + レコードデータ

ー Windows:

レコードデータ + STOP CODE(1バイト)

- ・「最大文字数/行」とは、YPS表記編集を用いて画面から入力できる、1行当たりの最大文字数のことです。半角文字は1バイト、全角文字は2バイトとして計算します。

また、システムによりコード系の内部表現方法が異なるため、各レコードに格納される実際のレコード長は、システムごとに異なります。

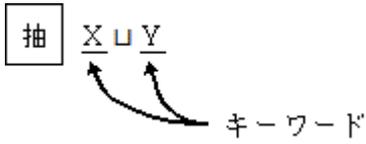
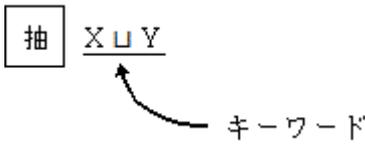
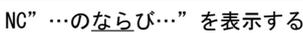
- ・ファイル形式とコード系は、仕様書の流通過程で、相手先のファイル形式とコード系に変換できます。例えば、WindowsのYPS表記編集で作成した仕様書をMSPへ転送する場合、ファイル転送時にMSPのファイル形式に変換するよう指定してください。Windowsへ転送する場合は、その逆を指定してください。

付録F 「FUJITSU YPS/COBOL言語文法書 1991年10月版」との相違

本文法書で示したYPS/COBOL言語の言語仕様は、「FUJITSU YPS/COBOL言語文法書 1991年10月版」以前の言語仕様と、「表F.1 「FUJITSU YPS/COBOL言語文法書 1991年10月版」との相違」に示す相違があります。表中の例において、「□」は空白を、「”」は全角の引用符を表しています。

相違となる箇所の検出方法、および相違となる言語仕様を含んだYPS仕様書の翻訳方法については、各システムの使用手引書を参照してください。

表F.1 「FUJITSU YPS/COBOL言語文法書 1991年10月版」との相違

項目	本文法書の言語仕様	旧文法書の言語仕様
YPSの識別子	30文字以内の文字列を指定します。 全角および半角の空白は区切り文字とみなされ ます。	文字列長は規定されていません(各システムの処 理系により規定されています)。 全角および半角の空白も識別子の一部とみなさ れます。
利用者定義の キーワード	30文字以内の文字列を指定します。 全角および半角の空白は区切り文字とみなされ ます。  例  「XY」、「X□Y」および「X□…□Y」のいずれでも引 用できます。	文字列長は規定されていません(各システムの処 理系により規定されています)。 全角および半角の空白もキーワードの一部とみな されます。  例  「X□Y」でしか引用できません
コメント記述文字 によるコメント	以下の場所に記述した記号「☆」は、コメント記述 文字とはみなされません。 ・ 文字列リテラル内 ・ 目的言語記述	コメント記述文字とみなされます。
文字列リテラル	文字列リテラル内は、すべてのYPSの機能の作用 対象外です。  例  " …の なら び…" を表示する 「なら」は標準コメントキーワードとはみなされませ ん。 全角文字記述機能を使用している場合に、文字 列リテラル内に全角の引用符を含めるときには、 文字列リテラルを囲う引用符を半角で記述します。	文字列リテラル内も、すべてのYPSの機能の作用 対象です。  例  " …の「 なら 」び…" を表示する 「なら」が標準コメントキーワードとみなされないよ うに、標準構文のエスケープ文字で囲う必要があ ります。 全角文字記述機能を使用している場合に、文字 列リテラル内に全角の引用符を含めるときには、 引用符を日本語記述のエスケープ文字で囲みま す。

項目	本文法書の言語仕様	旧文法書の言語仕様
	 例 <hr/> <div style="border: 1px solid black; padding: 2px;">NC "... " ... "を表示する</div> <hr/>	 例 <hr/> <div style="border: 1px solid black; padding: 2px;">NC " ... 《 》 ... " を表示する</div> <hr/>
COBOLオプションの並び	<p>プログラム開始図記号の本文に記述します。</p>  例 <hr/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">始</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">累積高の累計</div> [QUOTE]	<p>モジュール本体開始図記号の本文に記述します。</p>  例 <hr/> <div style="border: 1px solid black; padding: 2px; display: inline-block;">始</div> <QUOTE> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;">累積高の累計</div>
手続き部の節(詳細部) ^(註)	節と節の間には、手続き部の節を構成する表記を記述することはできません。	節と節の間にも、手続き部の節を構成する表記を記述することができます。
	節名は日本語名標宣言機能および全角文字記述機能の作用対象になります。	節名は日本語名標宣言機能および全角文字記述機能の作用対象になりません。
	節の有効範囲は、節を記述したプログラム単位または内部プログラム単位です。	節の有効範囲は下記のとおりです。 <ul style="list-style-type: none"> プログラム単位内に節を記述した場合 記述したプログラム単位 内部プログラム単位内に節を記述した場合 記述した内部プログラム単位を含むプログラム単位
	節には、直前の文から制御が渡ります。	節には、直前の文から制御は渡りません。
目的言語記述	目的言語記述の終わりには、丸図記号を記述しません。	目的言語記述の終わりには、丸図記号を記述します。
取込み機能	インクルード仕様書に記述できる表記を明確にしました。詳細は“5.4 インクルード仕様書”を参照してください。	インクルード仕様書に記述できる表記が明確に定義されていません。
	インクルード仕様書内に以下の表記を記述する場合には、開始と終了を対応させて記述します。 <ul style="list-style-type: none"> 抽象化処理構文定義 環境部、プログラム内データ部および手続き部の節 	開始と終了が対応していなくてもかまいません。
コメントキーワード定義機能	コメントキーワードの有効範囲は、コメントキーワード定義文を含む翻訳単位です。	コメントキーワードの有効範囲は、コメントキーワード定義文を記述した次の文から、モジュールの終わりまでです。  例 <hr/> 左記の例で、定義体内の語「ために、」はコメントキーワードとはみなされません。

項目	本文法書の言語仕様	旧文法書の言語仕様
	<p> 例</p> <hr style="border-top: 1px dotted #000080;"/> <p>○ 構文定義部</p> <p>□ 抽</p> <p>├── <u>…ために, …</u></p> <p>├── …</p> <p>□ 終</p> <p>└── 【ために,】を前置きのコメント表現として追加する</p> <p>定義体内の語「ために,」はコメントキーワードとみなされます。</p> <hr style="border-top: 1px dotted #000080;"/> <p>以下の表記の本文には、前置きコメントキーワードを用いてコメントを付加することはできません。</p> <ul style="list-style-type: none"> ・ 空文 ・ 本文がループ制御であるループ系の表記 <p>条件付き翻訳判定図記号、条件付き翻訳選択図記号および条件付き翻訳に従属する丸図記号の本文には、コメントキーワードを用いてコメントを付加することはできません。</p> <p>以下の表記の本文には、コメントキーワードを用いてコメントを付加することはできません。</p> <ul style="list-style-type: none"> ・ コメントキーワード定義文 ・ 一時変数宣言文 ・ 抽象化条件式定義文 ・ 日本語名標宣言文 	<p>前置きコメントキーワードを用いてコメントを付加することができます。</p> <p>コメントキーワードを用いてコメントを付加することができます。</p> <p>コメントキーワードを用いてコメントを付加することができます。</p>
構文定義機能	<p>定義体内に日本語名標宣言文を記述することはできません。</p> <p>抽象化条件式構文のキーワードに、「(」および「)」を含めることはできません。</p> <p>例外処理図記号(検索処理表記)に従属する丸図記号の本文に、抽象化条件式構文の引用文を記述することができます。</p> <p>条件式内で、「(」および「)」の内と外に跨がって、抽象化条件式構文を引用することはできません。</p>	<p>定義体内に日本語名標宣言文を記述することもできます。</p> <p>「(」および「)」を含めることもできます。</p> <p>抽象化条件式構文の引用文は記述できません。</p> <p>抽象化条件式構文を引用することはできません。</p>
日本語名標宣言機能	文字列1に特殊文字を含めることはできません。ただし、文字列1に1文字の特殊文字を指定することはできます。	全角文字記述機能を使用していなければ、文字列1に特殊文字を含めることができます。
全角文字記述機能	日本語名標宣言機能により置き換えられた文字列も、全角文字記述機能の変換対象になります。	日本語名標宣言機能により置き換えられた文字列は、全角文字記述機能の変換対象にはなりません。

注) 手続き部の節を、COBOLの節に対応する表記に変更しました。

付録G 翻訳指示行に記述する旧仕様に関する指示

翻訳指示行に、「言語仕様(1)」を指示した場合には、旧仕様の解釈に関する他の指示も記述することができます。

記述形式

◇翻訳：言語仕様（1，旧仕様の解釈に関する指示の並び）…

記述規則

旧仕様の解釈に関する指示の並びには、“表G.1 旧仕様の解釈に関する指示の一覧”で示した指示を、全角または半角の「,」で区切って記述します。

表G.1 旧仕様の解釈に関する指示の一覧

指示	意味
{ コメント継続あり コメント継続なし }	目的言語記述内に記述した、コメント記述文字によるコメントの認識方法を指示します。 <ul style="list-style-type: none"> コメント継続あり 目的言語記述のすべての継続行を一文とみなして、コメントの範囲を認識します。 コメント継続なし 目的言語記述の各継続行を一文とみなして、コメントの範囲を認識します。
{ 節整列あり 節整列なし }	COBOL原始プログラムを生成する際、環境部およびデータ部の各節の順序を、COBOL言語の仕様で定められた順序に並び換えるかどうかを指示します。 <ul style="list-style-type: none"> 節整列あり 並び換えを行います。 節整列なし 並び換えは行いません。
{ 日本語記述あり 日本語記述なし }	日本語名標宣言文と日本語記述のエスケープ文字の扱いを指示します。 <ul style="list-style-type: none"> 日本語記述あり 日本語名標宣言文および日本語記述のエスケープ文字として扱います。 日本語記述なし 日本語名標宣言文および日本語記述のエスケープ文字として扱いません。

記述例

◇翻訳：言語仕様（1，節整列あり，日本語記述なし），引用符（複）

用語集

後置きコメントキーワード

そのキーワードの文字列より後ろの文字列がコメントであることを示す文字列です。コメントキーワード定義文で定義します。

後判定ループ

条件式が真になるまで従属する処理を繰り返し実行する表記です。条件式の評価は、従属する処理を実行した後に行われます。

一時変数

定義体の中で一時的に使用する変数です。一時変数宣言文で宣言します。整数型、文字型、浮動小数点型の三つのデータ型を指定することができます。

一時変数宣言文

一時変数を宣言する表記です。定義体の先頭に記述します。

入口名

二次入口に付ける名前です。

インクルード仕様書

プログラムの一部が記述されたYPS仕様書です。取込み表記により取り込むことができます。

外部構文定義仕様書

複数のYPS仕様書で引用するコメントキーワード、抽象化処理構文および抽象化条件式構文を定義するYPS仕様書です。

外部日本語名標宣言仕様書

複数のYPS仕様書で引用する日本語名標を宣言するYPS仕様書です。

外部利用者定義構文

外部構文定義仕様書の構文定義部で定義した利用者定義構文です。

概要部

段階的詳細化では、節呼出しのことを概要部といいます。

[Win]

画面宣言節

プログラム内データ部を構成する節の一つで、ディスプレイに表示される画面のフォーマットを定義するところです。COBOLの画面節に対応する表記です。

仮引数宣言節

プログラム内データ部を構成する節の一つで、呼ぶプログラムから渡される変数を宣言するところです。COBOLの連絡節に対応する表記です。

環境部

翻訳時または実行時の計算機の環境、および実行時に使用するファイルの情報を記述するところです。

キーワード

抽象化処理構文および抽象化条件式構文において、構文を一意に識別するための文字列です。

キーワード群

空白列で区切られたキーワードの集まりです。

偽節

判定表記においては、条件式の評価結果が偽の場合に行う処理を記述するところです。

条件付き翻訳判定表記においては、翻訳条件式の評価結果が偽の場合に翻訳の対象とする表記を記述するところです。

基底変数宣言節

プログラム内データ部を構成する節の一つで、ポインタデータ項目により領域のアドレスが決定される変数を宣言するところです。COBOLの基底場所節に対応する表記です。

機能概要

モジュール属性記述行を構成する要素の一つで、モジュールの機能概要を表しています。

強制改ページ

YPS仕様書の印刷時に、強制的に改ページを行うことを指示する表記です。

空文

順次図記号の本文にシステム定義のキーワード「NONE」を記述した表記です。何の処理も行いません。

継続行

一つの表記の本文を2行以上に渡って記述した場合の2行目以降の行です。

検索処理

指定した条件を満足する表要素を検索し、対応する指標の値がその表要素を指すようにする表記です。COBOLのSEARCH文に対応します。

更新型ループ

ループ開始前に初期値を変数へ設定し、従属する処理を実行し、その後、毎回更新値を変数へ設定し、従属する処理を繰り返し実行する表記です。

更新型ループ制御

更新型ループの繰り返し条件を記述する式です。更復図記号の本文に記述します。

更新型ループと後判定ループの組み合わせ

更新型ループと後判定ループを組み合わせた表記です。

更新型ループと前判定ループの組み合わせ

更新型ループと前判定ループを組み合わせた表記です。

構造体

階層構造を持つデータを宣言する表記です。COBOLの集団項目に対応します。

構造体名

構造体の名前です。構造体図記号の本文に記述します。

構文定義機能

一連の処理や条件をプロジェクトの用語を使った日本語で記述するための機能です。抽象化処理構文定義機能と抽象化条件式定義機能があります。

構文定義のエスケープ文字

構文定義のエスケープ文字は、本文中の文字列をコメントキーワード定義機能および構文定義機能の作用対象から除外したいときに使用します。対象から除外したい文字列を構文定義のエスケープ文字「`「`」と`」`」で囲みます。

構文定義部

コメントキーワード、抽象化処理構文、抽象化条件式構文を定義するところです。

コメントキーワード

そのキーワードの文字列より前または後ろの文字列がコメントであることを示す文字列です。コメントキーワードには、前置きコメントキーワードと後置きコメントキーワードがあります。コメントキーワード定義文で定義します。

コメントキーワード定義機能

コメントキーワードを定義する機能です。コメントキーワード機能を用いることにより、本文中の処理や条件に自然な補足説明(コメント)を付加することができます。

コメントキーワード定義文

コメントキーワードを定義する表記です。構文定義部に記述します。

コメント記述文字

本文中の任意の位置にコメントを記述するための文字(「☆」)です。

コメント文

順次図記号の本文に、コメント記述文字によるコメントだけが記述されている表記です。

作業用データ宣言節

プログラム内データ部を構成する節の一つで、プログラム単位内、内部プログラム単位内で使用する変数を宣言するところです。COBOLの作業場所節に対応する表記です。

作成日付

モジュール属性記述行を構成する要素の一つで、モジュールの作成日付を表しています。

システム環境宣言節

環境部を構成する節の一つで、翻訳時および実行時の計算機の環境を宣言するところです。COBOLの構成節に対応する表記です。

システム定義のキーワード

YPS表記の記述形式として規定されている文字列および記号です。コメント記述文字、構文定義のエスケープ文字および日本語記述のエスケープ文字も、システム定義のキーワードです。本書の記述形式では、下線で示してあります。

指定回数ループ

指定した回数だけ従属する処理を繰り返し実行する表記です。反復図記号を使用して表現します。

指定回数型ループ制御

指定回数ループの繰り返し条件を記述する式です。反復図記号の本文に記述します。

終了条件前判定ループ

条件式が真になるまで従属する処理を繰り返し実行する表記です。条件式の評価は、従属する処理を実行する前に行われます。

主制御線

制御線のうち、最も左端にある制御線です。物理的な主制御線と論理的な主制御線があります。

順次文

記述された順に処理を実行する表記です。

条件式

評価の結果として真または偽の論理値を返す式です。

条件式定義体

抽象化条件式構文として定義したCOBOLの条件式です。

条件指定節

本文にシステム定義のキーワード(「真」、「偽」および「その他」)、発生事象以外の文字列を記述する選択節です。

条件付き翻訳機能

翻訳時のオプションにより、翻訳の対象にする表記を切り分ける機能です。

条件付き翻訳単一選択(選択条件名指定)

翻訳時のオプションで指定された選択条件名と条件指定節の本文に記述された選択条件名を上から比較し、最初に文字列が一致した条件指定節の表記を翻訳の対象にする表記です。

条件付き翻訳単一選択(翻訳条件式指定)

翻訳条件式を上から評価し、最初に翻訳条件式が真になった条件指定節の表記を翻訳の対象にする表記です。

条件付き翻訳判定

翻訳条件式を評価し、その評価結果により翻訳の対象とする表記を選択する表記です。

詳細部

段階的詳細化では、節のことを詳細部といいます。

真節

判定表記においては、条件式の評価結果が真の場合に行う処理を記述するところです。

条件付き翻訳判定表記においては、翻訳条件式の評価結果が真の場合に翻訳の対象とする表記を記述するところです。

図記号

YPS表記編集でYPS仕様書を作成する時に、メニューから選択する部分です。プログラムの制御構造を表現しています。

制御線

モジュール本体において、図記号と図記号を繋ぐ縦の線です。物理的な制御線と論理的な制御線があります。

制御線のネスト

ある表記を別の表記の中に記述することにより、制御線が多重になることをいいます。

制御線のネストレベル

制御線のネストの多重度です。主制御線を0とし、右にいくほど数字が大きくなります。YPS仕様書に記述できる物理的な制御線のネストレベルは10までです。

節

ひとまとまりの処理を記述するところです。COBOLの節に対応する表記です。節のことを詳細部ということもあります。

節出口

当表記が含まれる節の最後に制御を移す表記です。節出口のことを段階的詳細化出口ということもあります。

節呼出し

指定した節の処理を実行する表記です。COBOLの一つの節名を記述したとPERFORM文に対応します。節呼出しのことを概要部ということもあります。

全角文字

文字の外枠の高さと幅の大ききの比がほぼ1:1になっている文字のことです。

全角文字記述機能

COBOL言語では半角文字で記述しなければならない文字列を、全角文字で記述できるようにするための、全角文字から半角文字への変換機能です。

宣言部分

例外事象が発生した場合の処理を記述するところです。COBOLの宣言部分(DECLARATIVES)に対応する表記です。

宣言文

環境部の各節において環境を、プログラム内データ部の各節においてデータを宣言する表記です。順次図記号を使用して表現します。

選択系の表記

プログラムの基本制御構造の一つで、指定された条件に当てはまる処理だけを選択して実行する表記の総称です。選択系の表記には以下の種類があります。

- ・ 判定
- ・ 単一選択(定数指定)
- ・ 単一選択(条件式指定)
- ・ 複合選択(定数指定)
- ・ 複合選択(条件式指定)
- ・ 評価

選択条件名

翻訳条件式を構成する要素の一つです。

選択節

選択系の表記、例外処理および検索処理で、ある条件を満たしたときの処理を記述するところです。真節、偽節、その他節、条件指定節、例外処理節の総称です。

その他節

単一選択および評価表記においては、条件を満たす条件指定節が存在しなかったときの処理を記述するところです。

条件付き翻訳単一選択表記においては、条件を満たす条件指定節が存在しなかったときに翻訳の対象とする表記を記述するところです。

単一選択(条件式指定)

条件式を上から評価し、最初に条件式が真になった条件指定節の処理を実行する表記です。

単一選択(定数指定)

変数と定数を上から比較し、最初に値が一致した条件指定節の処理を実行する表記です。

段階的詳細化

節および節呼出しを使用してトップダウンプログラミングを行う方法です。

段階的詳細化処理名

段階的詳細化では、節名のことを段階的詳細化処理名といいます。

段階的詳細化出口

段階的詳細化では、節出口のことを段階的詳細化出口といいます。

担当者名

モジュール属性記述行を構成する要素の一つで、モジュールの作成担当者を表しています。

置換語句宣言

取込み表記の本文に記述する、取込み時に置き換える文字列を宣言する式です。

注釈

開始と終了で囲まれた表記を、翻訳の対象から無条件に除外する表記です。

抽象化条件式定義機能

COBOLの条件式に対して日本語の構文を定義する機能です。

抽象化条件式構文

抽象化条件式定義機能により、COBOLの条件式に対して定義した構文です。

抽象化条件式構文の引用式

抽象化条件式構文を引用する式です。単に引用式ということもあります。条件式という記述形式を持つYPS表記の本文に記述します。

抽象化条件式定義文

抽象化条件式構文を定義する表記です。構文定義部に記述します。

抽象化処理構文

抽象化処理構文定義機能により、一連の処理または宣言に対して定義した構文です。

抽象化処理構文定義

抽象化処理構文を定義する表記です。構文定義部に記述します。

抽象化処理構文定義機能

一連の処理または宣言に対して日本語の構文を定義する機能です。

抽象化処理構文の引用文

抽象化処理構文を引用する文です。単に引用文ということもあります。順次図記号または例外処理図記号を使用して表現します。

著作権表示行

YPS仕様書の著作権の所在を記述するための表記です。YPS仕様書の種類に関係なく、モジュール属性記述行の直後に記述します。

定義体

抽象化処理構文として定義した一連の処理または宣言です。

定数宣言節

プログラム内データ部を構成する節の一つで、プログラムの実行中に値が変わらない変数を宣言するところです。COBOLの定数節に対応する表記です。

手続き部

プログラムの処理手続きを記述するところです。宣言部分と手続き部分から構成されます。

特殊文字

取込み時の文字列置き換え、日本語名標宣言機能、全角文字記述機能において、対象の文字列を区切る文字になります。YPS仕様書に記述できる特殊文字とは異なります。

途中脱出

入れ子になったループ系の表記および選択系の表記の処理の途中で、主制御線上の処理に制御を移す表記です。

取込み

取込み機能において、インクルード仕様書を取り込む場所と、取り込むインクルード仕様書の名前を表す表記です。

取込み名

取込み表記の本文に記述する、取り込むインクルード仕様書の名前です。

取込み機能

プログラムの一部を他のYPS仕様書から取り込む機能です。取込み時に文字列または文字列の一部を他の文字列に置き換えることもできます。

内部プログラム単位

COBOLの含まれるプログラムに対応する表記です。内部プログラム開始から、対応する終了までの範囲をいいます。

内部プログラム呼出し

内部プログラム単位に制御を移す表記です。

内部利用者定義構文

プログラム仕様書の構文定義部で定義した利用者定義構文です。

二次入口

プログラムの主入口とは別のプログラムの入口を表す表記です。

日本語記述のエスケープ文字

日本語記述のエスケープ文字は、本文中の文字列を日本語名標宣言機能および全角文字記述機能の作用対象から除外したいときに使用します。対象から除外したい文字列を日本語記述のエスケープ文字「《》と「》」で囲みます。

日本語名標

日本語名標宣言機能において、COBOLの文字列に対して付けた日本語の名標です。日本語名標宣言文では、文字列1と書いています。

日本語名標宣言機能

本文中に記述するCOBOLの文字列(語、定数、予約語など)に対して、日本語の名標を付けるための、名標置き換え機能です。

日本語名標宣言文

日本語名標を宣言する表記です。

日本語論理演算子

YPSの文法により規定される演算子で、「かつ」と「または」があります。

入出力環境宣言節

環境部を構成する節の一つで、実行時に使用するファイルの情報を記述するところです。COBOLの入出力節に対応する表記です。

バージョンレベル

モジュール属性記述行を構成する要素の一つで、モジュールのバージョンレベルを表しています。

パラメタ

プログラム単位、内部プログラム単位および二次入口においては、プログラムに渡される変数のことです。

構文定義機能においては、構文を引用する場所により変更できる部分のことです。

パラメタ名

構文定義機能の可変部分を一意にするために付ける名前です。

半角文字

文字の外枠の幅が、全角文字の半分のサイズになっている文字です。

判定

条件式を判定し、その結果が真ならば真節の処理を実行し、偽ならば偽節の処理を実行する表記です。

評価

選択主体の組と選択対象の組を評価し、評価結果が真となる条件指定節の処理を実行する表記です。

表記

YPS表記のことを単に表記ということもあります。

標準構文

システムがあらかじめ用意した構文です。抽象化処理構文定義、抽象化条件式定義文で定義しないで使用することができます。標準構文には、YPS標準構文、YPS/COBOL標準構文、YPS/COBOL事務処理構文があります。

標準コメントキーワード

システムがあらかじめ用意したコメントキーワードです。コメントキーワード定義文で定義しないで使用することができます。

標準日本語名標

システムがあらかじめ用意した日本語名標です。日本語名標宣言文で宣言しないで使用することができます。標準日本語名標には、各システムで共通して使用できるSIA日本語名標と、各システムに固有な日本語名標とがあります。

ファイル形式宣言節

プログラム内データ部を構成する節の一つで、実行時に使用するファイルの構造を宣言するところです。COBOLのファイル節に対応する表記です。

複合選択(条件式指定)

条件式を上から評価し、条件式が真になった条件指定節の処理をすべて実行する表記です。

複合選択(定数指定)

変数と定数を上から比較し、値が一致した条件指定節の処理をすべて実行する表記です。

復帰

呼び出し元のプログラムに制御を戻す表記です。

物理的な制御線

YPS仕様書に記述された、見た目そのものの制御線です。

プログラム実行終了

プログラムの実行を終了する表記です。

プログラム仕様書

翻訳時に主な入力となるYPS仕様書です。

プログラム単位

COBOLの原始プログラムに対応する表記です。プログラム開始から、対応する終了までの範囲をいいます。

プログラム内データ部

プログラム単位、内部プログラム単位で使用するデータを宣言するところです。日本語名標を宣言することもできます。

プログラム名

プログラム単位および内部プログラム単位につける名前です。

プログラム呼出し

別のプログラムまたは二次入口に制御を移す表記です。

分岐

指定された分岐先に無条件に分岐する表記です。COBOLのGO TO文に対応します。

変数型ループ

変数が初期値より始まり、増分値ずつ増やしながらか、最終値より大きくなるまで、従属する処理を繰り返し実行する表記です。反復図記号を使用して表現します。

変数型ループ制御

変数型ループの繰り返し条件を記述する式です。反復図記号の本文に記述します。

変数型ループと後判定ループの組合せ

変数型ループと後判定ループを組み合わせた表記です。

変数型ループと終了条件前判定ループの組合せ

変数型ループと終了条件前判定ループを組み合わせた表記です。

変数型ループと前判定ループの組合せ

変数型ループと前判定ループを組み合わせた表記です。

報告書式宣言節

プログラム内データ部を構成する節の一つで、報告書ファイルに書き出される報告書の書式を宣言するところです。COBOLの報告書節に対応する表記です。

本文

YPS表記編集でYPS仕様書を作成する時に、図記号の右側または中に利用者が直接記述する部分です。

翻訳指示行

YPS仕様書の解釈に関する指示を記述する表記です。プログラム仕様書の著作権表示行の直後に記述します。

翻訳条件式

条件付き翻訳機能において、翻訳の対象にする条件を規定する式です。

翻訳条件名

翻訳条件式を構成する要素の一つです。

前置きコメントキーワード

そのキーワードの文字列より前の文字列がコメントであることを示す文字列です。コメントキーワード定義文で定義します。

前判定ループ

条件式が真の間、従属する処理を繰り返し実行する表記です。条件式の評価は、従属する処理を実行する前に行われます。

無限ループ

従属する処理を無限に繰り返し実行する表記です。反復図記号を使用して表現します。

目的言語記述

YPS仕様書内にCOBOLソースを直接記述するとき使用する表記です。

モジュール

YPS仕様書の別の呼びかたです。

モジュール概要記述

モジュールの機能やインターフェースなどの情報を記述するところです。YPS仕様書の種類に関係なく、モジュール本体の直前に記述します。

モジュール関連情報

YPS仕様書の機能概要やモジュール名など、モジュールに関する情報を記述するところです。モジュール属性記述行、著作権表示行、翻訳指示行およびモジュール概要記述で構成されます。

モジュール共通データ部

日本語名標を宣言するところです。

モジュール属性記述行

YPS仕様書の見出し情報であり、仕様書の先頭に記述します。モジュール名、機能概要、バージョンレベル、担当者名および作成日付の五つの情報で構成されます。

モジュール本体

データや環境の宣言および処理手続きなど、モジュールの実体を記述するところです。

モジュール名

モジュール属性記述行を構成する要素の一つで、モジュールの名前を表しています。

文字列リテラル

YPS/COBOL言語では、COBOLの定数のうち、引用符で囲って値を表すものを総称して文字列リテラルといいます。

要約定義

一連の処理や宣言に対して、その概要を説明するためのコメントを付加する表記です。

要約文

要約図記号の本文に記述するコメントです。

ライブラリ名

取込み図記号の本文に記述する、インクルード仕様書の格納場所の名前です。

ラベル

手続き部の段落の始まりを示す表記です。

ラベル名

ラベルの名前です。COBOLの段落名を記述します。

利用者定義構文

利用者が抽象化処理構文定義、抽象化条件式定義文で定義した構文です。

利用者定義のキーワード

構文定義機能およびコメントキーワード定義機能により、キーワードとして定義された文字列です。標準構文のキーワードおよび標準コメントキーワードも利用者定義のキーワードに含まれます。

ループ系の表記

プログラムの基本制御構造の一つで、制御条件の内容が満たされている間、図記号に従属する処理を繰り返し実行する表記の総称です。ループ系の表記には以下の種類があります。

- ・ 前判定ループ
- ・ 後判定ループ
- ・ 終了条件前判定ループ
- ・ 変数型ループ
- ・ 無限ループ
- ・ 指定回数ループ
- ・ 更新型ループ
- ・ 変数型ループと前判定ループの組合せ
- ・ 変数型ループと後判定ループの組合せ
- ・ 変数型ループと終了条件前判定ループの組合せ
- ・ 更新型ループと前判定ループの組合せ
- ・ 更新型ループと後判定ループの組合せ

ループ制御

ループの繰り返し条件を記述する式です。変数型ループ制御、指定回数型ループ制御、更新型ループ制御があります。反復図記号の本文に記述します。

例外処理

ある処理を行って特定の事象が発生した場合に、特定の処理を実行する表記です。

例外処理節

例外処理表記において、特定の事象が発生した場合の処理を記述するところです。

論理的な制御線

以下に示すYPSの機能およびYPS表記の解釈の結果、生成されるYPS仕様書の制御線です。

- YPSの機能
 - 条件付き翻訳機能による翻訳対象の取捨選択
 - 取込み機能によるインクルード仕様書の取込み
 - 抽象化処理構文定義機能における引用文の展開
- YPS表記
 - 要約定義表記の解釈

COBOLオプションの並び

COBOL原始プログラムの翻訳時にCOBOLコンパイラに与える指示です。プログラム開始図記号の本文に記述します。

COBOLの文字列

COBOLの文法により規定される文字列です。

YPS仕様書

YPS/COBOL言語の文法に基づいて記述されたプログラムのことです。プログラム仕様書、インクルード仕様書、外部構文定義仕様書および外部日本語名標宣言仕様書があります。

YPS仕様書のことをモジュールということもあります。

YPS表記

YPS仕様書を構成する単位です。図記号と本文の組合せであれば、どんなに小さな単位でもYPS表記です。また、複数のYPS表記が組み合わさって、大きなYPS表記を構成します。YPS表記のことを単に表記ということもあります。

YPSの演算子

YPSの文法により規定される演算子です。日本語論理演算子があります。

YPSの機能

以下に示す機能の総称です。

- 条件付き翻訳機能
- 取込み機能
- コメントキーワード定義機能
- 構文定義機能
- 日本語名標宣言機能
- 全角文字記述機能

YPSのキーワード

YPSの文法により規定される特定の文字列です。システム定義のキーワードと利用者定義のキーワードに分類されます。

YPSの区切り文字

YPSの文法により規定される部分で、本文の要素を区切るための文字です。以下の文字があります。

- モジュール属性記述行における全角または半角の「,」または全角の「,」
- 翻訳指示行における全角または半角の「,」
- モジュール本体内における全角または半角の空白列

YPSのコメント

プログラムの意味に何の影響も与えない文字列です。以下の種類があります。

- モジュール関連情報内
- モジュール属性記述内の各項目
- 著作権表示
- モジュール概要記述
- モジュール本体内容
- コメント記述文字によるコメント
- 要約定義によるコメント
- コメントキーワードによるコメント

YPSの識別子

YPSの機能を構成する要素に対して利用者が付けた名前です。以下に示すものがあります。

- 条件付き翻訳機能の翻訳条件名および選択条件名
- 取込み機能の取込み名およびライブラリ名
- 置換語句宣言の語1～語4
- 抽象化処理構文定義機能および抽象化条件式定義機能のパラメタ名
- 抽象化処理構文定義機能の一時変数名
- 日本語名標宣言機能の文字列1

YPSの文字列

YPS仕様書の本文に記述する文字列のうち、YPSの文法により規定される文字列です。YPSの文字列には以下の種類があります。

- YPSのキーワード
- YPSの識別子
- YPSの演算子
- YPSのコメント
- YPSの区切り文字

YPS標準構文

YPSの各言語に共通する標準構文です。

YPS/COBOL標準構文

YPS/COBOL言語の標準構文です。

YPS/COBOL事務処理構文

事務処理に関する構文の集まりです。標準日本語名標と組み合わせて使用します。

索引

[A]	YPS表記の構成..... 8
AT END指定を行うFIND文、GET文またはSTORE文..... 154	YPS標準構文..... 120,156,192
[C]	[あ]
COBOLオプションの並び..... 29,191	後置きコメントキーワード..... 117,180
COBOL言語との関係..... 1	後判定ループ..... 63,180
COBOL言語の環境部..... 141	後復号記号..... 64
COBOL言語の手続き部..... 144	一時変数..... 121,122,180
COBOL言語のデータ部..... 142	一時変数宣言文..... 121,122,180
COBOL言語のプログラム終わり見出し..... 141	一時変数名..... 123
COBOL言語の翻訳単位..... 140	入口名..... 91,180
COBOL言語の見出し部..... 141	インクルード仕様書..... 6,110,115,180
COBOLの条件式..... 129	引用式..... 130
COBOLの文..... 47,52	引用文..... 123
COBOLの文字列..... 133,137,191	引用文と一致する抽象化処理構文の検索方法..... 124
COPY文およびREPLACE文で操作できないCOBOLの文..... 150	埋込みSQLの記述..... 153
COPY文およびREPLACE文の終止符の記述..... 153	英小文字で記述することができないCOBOLの予約語..... 153
[E]	置換えの手順..... 113,134
ENTRY文..... 90	同じ抽象化処理構文の優先順位..... 124
EVALUATE文..... 60	[か]
[G]	解釈および結合順序の変更..... 129,130
GO TO文..... 92	外部構文定義仕様書..... 6,131,180
[M]	外部日本語名標宣言仕様書..... 6,135,180
Mシリーズ固有日本語名標..... 169	外部利用者定義構文..... 120,180
[N]	概要部..... 139,180
NEXT SENTENCE指定を行うIF文およびSEARCH文..... 155	画面節..... 45
[S]	画面宣言節..... 33,45,180
SEARCH文..... 86	仮引数宣言節..... 33,43,180
SIA共通文法..... 1,2	環境部..... 30,31,141,180
SIA日本語名標..... 165	環境部の節を構成する表記..... 39
[U]	環境部を構成する表記..... 37
USING指定..... 141	記述文字..... 19
[Y]	偽節..... 53,106,181
YPS..... 1	基底場所節..... 39
YPS/COBOL事務処理構文..... 120	基底変数宣言節..... 33,39,181
YPS/COBOL言語..... 1	機能概要..... 25,181
YPS/COBOL事務処理構文..... 158,192	強制改ページ..... 101,181
YPS/COBOL標準構文..... 120,156,192	キーワード..... 122,128,180
YPSコンパイラが生成する段落の見出し..... 153	キーワード群..... 124,180
YPS仕様書..... 1,3,191	空文..... 52,181
YPS仕様書を構成する表記..... 23	継続行..... 20,181
YPSの演算子..... 191	結合順序の変更..... 104,105
YPSの機能..... 191	検索処理..... 86,181
YPSのキーワード..... 16,191	語1..... 113
YPSの区切り文字..... 19,191	語2..... 113
YPSのコメント..... 17,191	語3..... 113
YPSの識別子..... 17,192	語4..... 113
YPSの文字列..... 16,192	更新型ループ..... 70,181
YPSの演算子..... 17	更新型ループ制御..... 71,181
YPS表記..... 7,191	更新型ループと後判定ループの組合せ..... 82
	更新型ループと後判定ループの組み合わせ..... 181
	更新型ループと前判定ループの組合せ..... 81
	更新型ループと前判定ループの組み合わせ..... 181
	更新値..... 71
	構成節..... 37

構造体.....	48,181
構造体図記号.....	48
構造体名.....	48,181
更復図記号.....	71
構文定義機能.....	20,120,130,181
構文定義機能のキーワード.....	114
構文定義のエスケープ文字.....	16,119,130,181
構文定義部.....	24,27,121,182
コメント記述文字.....	16,182
コメント記述文字によるコメント.....	18,100,117
コメントキーワード.....	27,117,182
コメントキーワード定義機能.....	20,117,130,182
コメントキーワード定義文.....	117,182
コメントキーワードによるコメント.....	18,117
コメントキーワードの引用.....	117
コメントの記述方法.....	18
コメント文.....	99,182

[さ]

最終値.....	67
作業場所節.....	41
作業用データ宣言節.....	33,41,182
作成日付.....	25,182
システム環境宣言節.....	31,37,182
システム固有日本語名標.....	165
システム固有文法.....	1,2
システム定義のキーワード.....	16,182
システムに固有なYPS表記.....	2
システムに固有な標準日本語名標.....	2
実行終了図記号.....	96
指定回数型ループ制御.....	69,182
指定回数ループ.....	69,182
集団項目.....	48
終復図記号.....	65
終了条件前判定ループ.....	64,182
主制御線.....	10,182
順次系の表記.....	51
順次図記号.....	47
順次文.....	51,182
条件式.....	53,128,130,183
条件式定義体.....	183
条件式定義文.....	127,128
条件指定節.....	54,56,57,59,60,87,107,108
条件指定節.....	183
条件付き翻訳機能.....	20,103,183
条件付き翻訳選択図記号.....	107
条件付き翻訳単一選択(選択条件名指定).....	103,107,183
条件付き翻訳単一選択(翻訳条件式指定).....	103,108,183
条件付き翻訳判定.....	103,105,183
条件付き翻訳判定図記号.....	106
詳細部.....	139,183
初期値.....	67,71
真節.....	53,106,183
図記号.....	8,171,183
制御線.....	183
制御線のネスト.....	10,183
制御線のネストレベル.....	10,183

節.....	50,139,183
節出口.....	95,183
節名.....	50,94
節呼出し.....	51,94,139,184
節呼出し図記号.....	94
全角文字.....	3,184
全角文字記述機能.....	21,135,137,184
宣言部分.....	35,49,184
宣言文.....	47,184
選択系の表記.....	52,93,184
選択主体の組.....	60
選択条件名.....	105,184
選択図記号.....	54
選択節.....	184
選択対象の組.....	60
選択節に記述できる表記.....	104
増分値.....	67
属性.....	29,36,48
そとPERFORM文.....	94
その他節.....	54,56,60,107,108,184

[た]

単一選択(条件式指定).....	55,184
単一選択(定数指定).....	54,184
段階的詳細化.....	139,184
段階的詳細化処理名.....	139,185
段階的詳細化出口.....	139,185
担当者名.....	25,185
段落.....	91
置換語句宣言.....	110,112,185
注釈.....	97,185
抽象化条件式構文.....	27,127,128,185
抽象化条件式構文の引用式.....	130,185
抽象化条件式定義機能.....	120,127,185
抽象化条件式定義文.....	127,185
抽象化条件式の引用式.....	129
抽象化処理構文.....	27,120,122,185
抽象化処理構文定義.....	120,121,185
抽象化処理構文定義機能.....	120,185
抽象化処理構文の引用文.....	123,185
著作権表示.....	18
著作権表示行.....	23,25,185
定義体.....	120,122,185
定数.....	19,54,58,70
定数節.....	42
定数宣言節.....	33,42,185
手続き部.....	30,34,144,186
手続き部の節を構成する表記.....	51
手続き部分.....	35
手続き部を構成する表記.....	49
データ記述項のレベル番号付け.....	143
データ部の段落の記述.....	143
特殊文字.....	113,134,137,186
途中脱出.....	93,186
取込み.....	110,111,186
取込み機能.....	20,110,186
取込み名.....	112,186

[な]

内部プログラム開始	141
内部プログラム終了	141
内部プログラム単位	30,35,186
内部プログラム単位を構成する表記	97
内部プログラム呼出し	89,186
内部プログラム呼出し図記号	89
内部利用者定義構文	120,186
二次入口	88,90,186
二次入口図記号	91
日本語記述のエスケープ文字	16,134,135,137,186
日本語名標	186
日本語名標宣言機能	20,133,135,186
日本語名標宣言文	28,133,186
日本語名標の置換え	134
日本語論理演算子	17,129,186
日本語論理演算子による結合	129
入出力環境宣言節	31,38,187

[は]

発生事象	84,87
パラメタ	29,35,88,90,91,122,128,187
パラメタ名	122,187
半角文字	3,187
判定	53,187
判定図記号	53
反復図記号	66
バージョンレベル	25,187
被検索処理文	86
評価	60,187
評価図記号	60
表記	7,187
標準構文	120,156,187
標準構文の種類	120
標準コメントキーワード	117,118,187
標準日本語名標	165,187
標準の後置きコメントキーワード	118
標準の前置きコメントキーワード	118
標準のライブラリ	112
被例外処理文	84
被例外処理文に記述できるCOBOLの文と発生事象の組合せ	85
ファイル形式宣言節	33,40,187
ファイル節	40
複合選択(条件式指定)	58,187
複合選択(定数指定)	57,187
複選図記号	57
復帰	95,187
復帰図記号	95
復帰値	95,96
物理的な主制御線	10
物理的な制御線	9,188
プログラム終わり見出し	141
プログラム開始	141
プログラム実行終了	96,188
プログラム終了	141
プログラム仕様書	6,188
プログラム単位	24,28,140,188

プログラム単位を構成する表記	30
プログラム内データ部	30,32,142,188
プログラム内データ部の各節を構成する表記	46
プログラム内データ部を構成する表記	39
プログラム名	29,35,88,90,188
プログラム呼出し	88,188
プログラム呼出し図記号	88
分岐	92,188
分岐先	93
分岐図記号	92
変数	54,57,67,70,71
変数型ループ	65,188
変数型ループ制御	66,76,79,188
変数型ループと後判定ループの組合せ	75,188
変数型ループと終了条件前判定ループの組合せ	78,188
変数型ループと前判定ループの組合せ	72,188
報告書式宣言節	33,44,188
報告書節	44
本文	8,15,188
翻訳オプションの指定方法	104
翻訳指示行	23,25,188
翻訳指示行に記述する指示	26
翻訳条件式	104,106,189
翻訳条件式の評価方法	105
翻訳条件名	104,107,189

[ま]

前置きコメントキーワード	117,189
前判定ループ	62,189
前復図記号	63
見出し部	141
見出し部の段落	141
無限ループ	68,189
目的言語記述	100,137,189
モジュール	3,189
モジュール概要記述	18,23,27,189
モジュール関連情報	6,23,115,131,136,189
モジュール関連情報の表記	7
モジュール関連情報を構成する表記	24
モジュール共通データ部	24,28,189
モジュール属性記述行	23,24,189
モジュール本体	6,23,115,131,136,189
モジュール本体の表記	7
モジュール本体を構成する表記	27
モジュール名	24,189
文字列1	134
文字列2	134
文字列の置換え	113
文字列リテラル	19,137,189
文字列リテラルの記述規則	19

[や]

要約定義	98,189
要約定義によるコメント	18
要約文	99,189

[ら]

ライブラリ名	112,190
--------	---------

ラベル.....	91,190
ラベル図記号.....	92
ラベル名.....	92,190
利用者定義構文.....	120,190
利用者定義のキーワード.....	16,114,190
利用者定義のコメントキーワード.....	117
ループ系の表記.....	62,93,190
ループ制御.....	190
例外事象.....	49
例外処理.....	84,190
例外処理図記号.....	84,86
例外処理節.....	84,87,190
連絡節.....	43
論理演算子による結合.....	104,105
論理的な主制御線.....	10
論理的な制御線.....	9,190