# 「スクール COBOL2002」の新機能の概要

「スクール COBOL2002」は、2002 年に ISO で制定された
第 4 次国際規格〔COBOL2002 規格〕の主要な機能をサポートした製品です。

## 1.翻訳指令による条件翻訳

「翻訳指令」が記述された箇所は、コンパイルの最初の段階でソースプログラムに対して、「翻訳指令」の種類に応じた処理を行います。「翻訳指令」のうち、「条件翻訳」は、値とソースプログラムの範囲を指定して記述します。「翻訳指令」で指定した値が、コンパイラオプションに設定された場合に、該当するソースプログラムの範囲をコンパイル対象にします。
この機能により、プラットフォーム固有の処理、バージョン固有の処理などが、1つのソースプログラム内で管理できます。

【記述例】

```
>>IF 翻訳変数V IS DEFINED
*>翻訳変数Vが定義されているときに有効となる
DISPLAY "DEFINED".
>>ELSE
*>翻訳変数Vが定義されていないときに有効となる
DISPLAY "NOT DEFINED".
>>END-IF
```

【その他の翻訳指令】

- 旧規格仕様警告（FLAG-85）
- ソースリスト出力指示（LISTING）
- 例外の伝播（PROPAGATE）
- ソース形式(自由/固定書式)(SOURCE)
- 発生する例外の指示（TURN）

など

## 2.TYPEDEF 句／SAME AS 句

「TYPEDEF 句」はデータのひな型を宣言できます。複数のデータで構成された「データ構造」を1つのデータ型として宣言し、「TYPE 句」で利用してデータの実体を定義することができます。
「SAME AS 句」は既に記述されているデータ構造と同じデータ構造を簡単に記述できます。
データ定義部分を簡潔に記述できます。

【記述例：TYPEDEF 句】

```
*>データ型「日付」を宣言する
01  日付  IS TYPEDEF.
  05  年    PIC 9(4).
  05  月    PIC 9(2).
  05  日    PIC 9(2).

*>データ型「日付」の「受注日」を定義する
01  受注日   TYPE 日付.
*>データ型「日付」の「納入日」を定義する
01  納入日   TYPE 日付.
```

## 3.利用者定義関数

COBOL が提供している組み込み関数に加えて、利用者が関数を作成できます。従来の入れ子プログラムとは異なり、IF 文などの条件式や計算式の中に記述でき、 判定内容や計算内容を判りやすく簡潔にまとめることができます。

【記述例:関数の定義】

```
IDENTIFICATION DIVISION.
FUNCTION-ID.  最大公約数.
DATA DIVISION.
LINKAGE SECTION.
 01  A  PIC S9(9).
 01  B  PIC S9(9).
 01  R  PIC S9(9).
PROCEDURE DIVISION
   USING A B RETURNING R.
...
```

【記述例:関数の使用(計算式)】

```
  COMPUTE 結果 = FUNCTION 最大公約数(X Y).
```

【記述例:関数の使用(条件式)】

```
  IF ( FUNCTION 最大公約数(X Y) < 10 ) THEN
  ...
```

## 4.共通例外処理

プログラム実行中に発生する様々なエラー要因が、例外名として細かく定められました。

従来からある、入出力エラー処理における USE 文の宣言手続きに加え、例外名を使うことでより柔軟な例外処理を記述できます。

【記述例:桁あふれの例外検出】

```
01  A  PIC 99.
...
PROCEDURE DIVISION.
DECLARATIVES.
 SIZE-OVERFLOW SECTION.
 USE AFTER EXCEPTION
   CONDITION EC-SIZE-OVERFLOW.
...
END DECLARATIVES.
>>TURN EC-SIZE-OVERFLOW CHECKING ON
  MOVE 100 TO A.
  *> システムが例外発生を自動検出
...
  IF A  >  60 THEN
    RAISE EXCEPTION  EC-SIZE-OVERFLOW
    *> 例外処理を任意に呼び出す
```

## 5.オブジェクト指向機能

C++や Java™と同様にオブジェクト指向プログラミングがサポートされました。

「継承」、「ポリモルフィズム」、「カプセル化」などの概念により、COBOL の持つ高い生産性をさらに高めることができます。

【記述例：クラス定義】

```
IDENTIFICATION DIVISION.
CLASS-ID.  当座 INHERITS 預金.
...
IDENTIFICATION DIVISION.
  OBJECT.
  DATA DIVISION.
  WORKING-STORAGE SECTION.
  01 残高 PIC S9(9).
  PROCEDURE DIVISION.
  IDENTIFICATION DIVISION.
    METHOD-ID. 預け入れ.
    ...
    END METHOD 預け入れ.
  END OBJECT.
END CLASS 当座.
```

インスタンス定義

メソッド定義

## 6.自由形式正書法

従来から規定されている「固定形式正書法」に加えて、行番号領域や標識領域のない「自由形式正書法」が採用されました。

【記述例：自由形式正書法】

```
L+++++++++1+++++++++2+++++++++3+++++++++4...
>>SOURCE FORMAT IS FREE
*> カラム位置を意識しないコーディング
口座入金 SECTION.
  ADD  財布  TO  口座.  *> 貯金が増える
```