

# 社会を支える“ことば” COBOL

2010年4月16日

**COBOLコンソーシアム 会長**

**ISO/IEC JTC 1/SC 22/WG 4 (COBOL) 主査**

**情報処理学会 情報規格調査会 SC 22/COBOL WG 主査**

**高木 渉 (日立)**

# 1. COBOL誕生のころ

# 1-1 COBOLはわずか1年で作られた

## ◆ 1959年4月8日

- 共通 業務用 言語 の必要性についての集まり

## ◆ 1959年6月23-24日

- Short-Range委員会初回会議

## ◆ 1960年1月7-8日

- Short-Range委員会の仕様をCODASYL承認  
(CODASYL: COncference on DAta SYstems Languages)

## ◆ 1960年4月

- 仕様書発行 (米国Government Printing Office)

(参考: Jean E. Sammet, The Early History of COBOL, History of Programming Languages, ACM, 1978.)

## 1-2 COBOL仕様開発時の方針

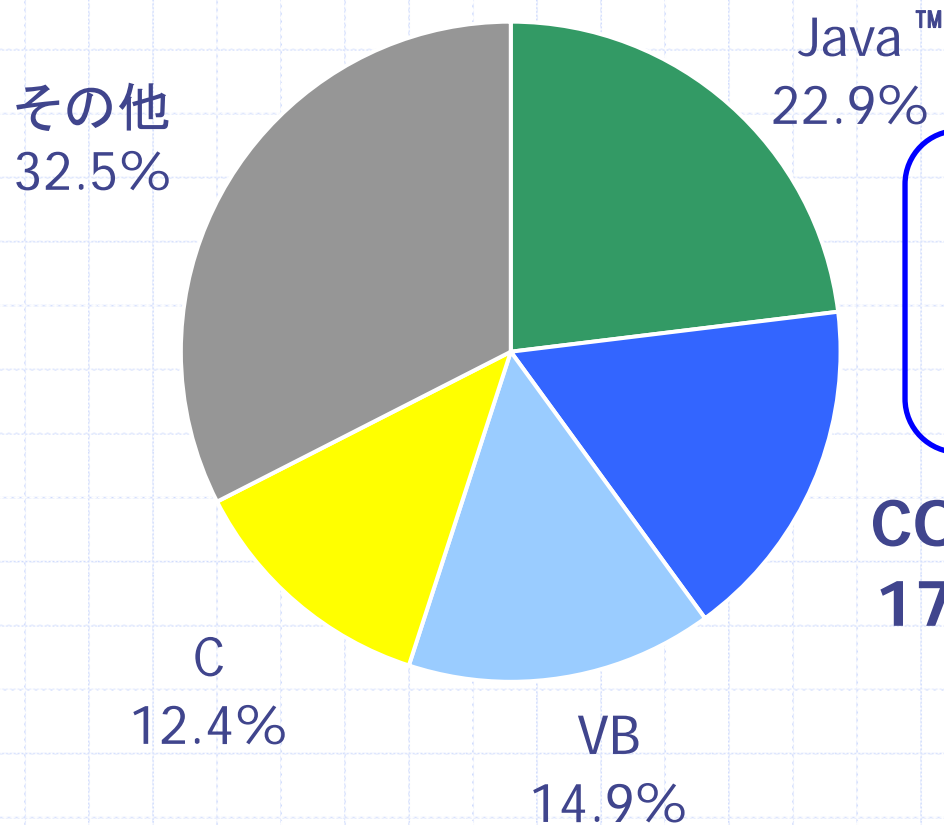
- ◆ 業務用データ処理向け
- ◆ 言語が計算機に独立 (移植性がある)
  - 移植性は1950年代から課題とされていた
- ◆ エレガントさより, 使いやすさ・読みやすさ
  - 書くことより読むこと
- ◆ 簡単な英語で書く
  - 経営側の人でも読める

## 1-3 COBOL-60のプログラム言語への功績

- ◆ 英語に近い表現で, さらに補助語を導入  
例: READ ファイル名 **RECORD** INTO 変数
- ◆ データ部と手続き部を分離
- ◆ 多段に階層化したレコード定義
- ◆ PICTURE文字列によるデータの定義
  - 当時参考にした言語から採用
- ◆ 変数名は30文字まで
- ◆ 数字項目は十進数の18桁まで

## 2. 社会に浸透しているCOBOL

## 2-1 開発言語の使用状況(日本国内)



新規開発で  
400万行以上は  
COBOLだけ

回答数: 2161/2327

出典: 日経BP社 ソフトウェア開発データ白書2009

## 2-2 社会基盤を支えるCOBOL

- ◆ 大規模から中小規模まで，生活に浸透
  - さまざま行政システム
  - さまざま企業内のシステム
  - 銀行オンラインシステム
  - 座席予約システム
- ◆ すでにインフラとして実際に稼動
  - 目立たないところで，役に立っている



### 3. COBOLの良さ

## 3-1 COBOLの特徴 (1/2)

### ◆レコード入出力オリエンテッド

- 「レコード」が入出力の単位
- フィールドのバイト位置は計算機のメモリ境界にとらわれない

### ◆型の概念の絶妙な弱さ

- メモリ範囲にレコード定義を重ねるだけで各フィールドが使えるようになる
- 型が違えばMOVE文はデータを型変換する

## 3-2 COBOLの特徴 (2/2)

### ◆ 静的言語

- メモリ設計が容易
- 読解時に注目すべきステートが少ない

### ◆ 一つの文に多くの処理を詰め込めない

- 読む速度で理解する

### ◆ 十進数オリエンテッド

- 計算機は二進数でも、人は十進数で生活する

### ◆ 長期にわたって互換性が保てられている

## 3-3 COBOLは長期稼動に有利

### ◆長い目で測る生産性

- 初期開発時
- 変更を繰り返したソースの修正・拡張時

### ◆修正と拡張に必要な要素

- 変更対象のプログラムを理解する
- 変更方法を考案する
- 変更が正しいことを確認する

変更箇所の正しさ + 非変更箇所の保存

## 4. 国際規格の動向

## 4-1 関連する標準化団体

### ◆ ISO/IEC JTC 1/SC 22/WG 4

- 国際標準化団体
- 大きな方針を決定
- 日本, 米国, 英国, オランダ, カナダ

### ◆ ISO/IEC JTC 1/SC 22/WG 4/OWG-1

- 実際の規格文面を作成

### ◆ 米国INCITS PL22.4 (OWG-1と実質同一)

- 実際の規格文面を作成

## 4-2 次期規格について

### ◆ 日程

- 2011年内の規格発行

### ◆ 既存機能のオプション化あり

- 画面入出力機能, ロケールなど

### ◆ 主な追加機能

- IEEE754の十進/二進浮動小数点数の扱い
- ANY LENGTH基本項目
- 容量可変表 (既存の可変長項目より柔軟)

## 5. これからのCOBOL



## 5-1 マイナスイメージを超えて

- ◆ 「古い」=「置き換えるべき」は間違い
  - 新旧は優劣につながらない
  - 使えているものを置き換える理由はない
- ◆ COBOLプログラマーはすぐに養成できる
  - COBOLの習得は難しくない
- ◆ 言語仕様の面白さは2番目の優先度で
  - 言語に恋するのではなく、業務と向き合おう

## 5-2 社会基盤は今後もCOBOLで

- ◆ COBOLは、社会基盤であり続ける
- ◆ COBOLは、長期稼動に向く数少ない言語

COBOL技術者は社会を支えています

「COBOL技術者」

から

「COBOLも使える技術者」へ



END

・Java 及びすべてのJava関連の商標及びロゴは、米国及びその他のにおける米国Sun Microsystems, Inc.の商標または登録商標です。