

スクールCOBOLユーザーの方のための情報誌

スクール COBOL Report

●巻頭寄稿

日本語とプログラム言語

東京国際大学言語コミュニケーション学部 教授
今城哲二

●COBOL最新企画

COBOL2002の便利な機能 —カウント情報

●ユーザー校事例

埼玉県立狭山経済高等学校
栗原一也

長崎市立長崎商業高等学校
杉原利幸

●企業におけるCOBOL活用

医薬品卸業界における 情報システムのオープン化

●「日立スクールCOBOL2002」を
使いこなそう!

【第8回】

サブプログラム構成の

プロジェクトファイルの作成手順

Vol. 9
2005.11

情報処理教育用COBOLコンパイラの決定版

●Windows®対応

スクールCOBOL2002シリーズ

HITACHI
Inspire the Next

使いなれた
COBOLで
GUI構築技法を
効率よく学習

・イベント手続きをCOBOLで記述
・COBOL専用エディタでソース編集

・部品パレットからGUI部品をドラッグ
& ドロップで貼り付け

スクールCOBOL2002 GUI構築バック画面

情報処理教育に最適。

高信頼性と豊富な実績のある「スクールCOBOL85」の後継製品、
「スクールCOBOL2002シリーズ」です。

操作性のよいWindows®に完全対応した高性能コンパイラと、今までにない効率的な実習環境を提供。

スクールCOBOL2002

- 最新の第4次国際規格に対応しています。
- 日本語メッセージを表示するなど、エラー修正も容易です。
- エラーメッセージをクリックすると、自動的にソースプログラムの該当行にカーソルが位置付きます。
- プログラムを実行しながら、どの命令を実行しているか画面に表示するデバッグツールを内蔵しているため、デバッグ作業も容易です。

スクールCOBOL2002GUI構築パック

- イベント駆動型GUIアプリケーションの学習ができます。
- スクールCOBOL2002の機能は、そのまま使えます。
- 部品パレットからGUI部品をドラッグ&ドロップするだけでGUI画面が作成できます。
- GUIで発生するイベントに対する手続きをCOBOLで記述できます。

※GUI:Graphical User Interface

価格(税込)

スクールCOBOL2002		スクールCOBOL2002 GUI構築パック	
● 1セット	60,900円	● 1セット	92,400円
● 21セット	987,000円	● 21セット	1,478,400円
● 41セット	1,811,250円	● 41セット	2,713,200円

※対応OS: Windows®2000、WindowsNT®4.0、Windows®Millennium Edition、Windows®98、Windows®XP
※Microsoft、Windows、WindowsNTは、米国およびその他の国における米国Microsoft Corp.の登録商標です。

販売 実教出版株式会社 〒102-8377 東京都千代田区五番町5
本社/TEL(03)3238-7777 FAX(03)3238-7755 大阪/TEL(06)6397-2400 FAX(06)6397-2402 九州/TEL(092)473-1841 FAX(092)471-7529
開発元 株式会社 日立製作所 ソフトウェア事業部

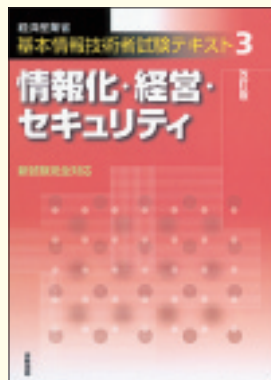


本カタログは環境に配慮し、植物性大豆油インキを使用しています。



古紙配合率100%再生紙を使用しています

◇基本情報技術者試験対策に◇



— 経済産業省基本情報技術者試験テキストシリーズ —

1. **ハードウェア・ソフトウェア** 三訂版
B5判 272p. 定価 2,000円

2. **システムの開発と運用** 三訂版
B5判 166p. 定価 1,600円

3. **情報化・経営・セキュリティ** 改訂版
B5判 144p. 定価 1,500円

4. **COBOL** B5判 288p. 定価 1,900円

5. **COBOL問題集** 改訂版
B5判 256p. 定価 1,900円



合格のための総仕上げ
2006年度版 **速攻基本情報技術者 午前**
A5判 216p. 定価 2,205円

合格のための総仕上げ
2006年度版 **速攻基本情報技術者 午後**
A5判 176p. 定価 2,048円

◇初級システムアドミニストレータ試験対策に◇

2006年度版
— 合格のための総仕上げ —
速攻 シ ス ア ド
A5判 256p. 定価 2,415円

チャレンジライセンス
初級システムアドミニストレータ
12月発行 予定 B5判 272p. 定価 1,800円

シスアドテキスト1
エンドユーザーコンピューティング
B5判 216p. 定価 1,880円

シスアドテキスト2
システム環境の管理・運用
B5判 192p. 定価 1,850円

* 定価はすべて5% 税込みです

実教出版株式会社

<http://www.jikkyo.co.jp/>

スクール COBOL 2002 の情報提供中!

本 社 千代田区五番町5
TEL 03(3238)7777 FAX 03(3238)7755
大 阪 大阪市淀川区宮原5-1-3 新大阪生島ビル
TEL 06(6397)2400 FAX 06(6397)2402
九 州 福岡市博多区博多駅前3-10-24 藤井ビル
TEL 092(473)1841 FAX 092(471)7529

日本語と プログラム言語



東京国際大学言語コミュニケーション学部 教授

今城哲二 Tetsuji Imajo

35年間、日立製作所で基本ソフトウェアの開発とマーケティングに従事。2004年から、東京国際大学。専門は、プログラム言語、コンパイラ、ソフトウェア工学。国際規格COBOL作成委員会の日本代表委員、日本工業規格COBOL原案作成委員会委員長など多くの標準化委員を歴任。COBOLコンソーシアム会長。博士（工学）。

日本語による教育の成功とカタカナ語の氾濫

1853年のペリー来航による開国を契機に、西洋文明が日本に押し寄せてきた。特に、明治維新以降、日本は積極的に西洋の知識を取り入れ、教育に力を注ぎ、西洋に追いつくことに専念した。高等教育においては、多くの外国人教師を招き、外国語で講義を行ったが、早い段階で日本人教師に切り替え、日本語による教育に転換した。このとき、多くの学術用語を外国語のままでなく日本語化したことが、日本語による教育の成功の大きな要因だった。

例えば、数学では、代数、幾何、微分、積分、面積、体積、平方、立法、群、環、体、集合、確率、統計など基本的な用語はすべて日本語化されている。また、“1たす1は2”というように、日本語で算数教育をしている。開国以来150年間ずっと、これを英語で“One plus one equals two.”と教え続けたとすれば、今の日本の教育普及率、文化水準、経済水準はとても達成できなかったことはだれもが分かることである。

コンピュータ用語については、音楽のスコアを楽譜というようにプログラムを算譜と名付けるなど、1970年代はじめるにカタカナ語からの脱却を図った提案があったが成功せず、今や多くの用語がカタカナ語で使われるという悲惨な状況にある。これは日本人の日本語造語力が明治時代とくらべ激減していることと、わからないカタカナ語を使うほうが先進的な感じがするという風潮が主要因であろう。

政府もカタカナ語の氾濫を憂慮している。2003年から4回にわたり国立国語研究所が、使用頻度が高いが一般の多

くが理解できないカタカナ語（外来語）を日本語に言い換える提案をしている¹⁾。すでに提案されているのは、デフォルトを債務不履行または初期設定、コンプライアンスを法令遵守など143語である。今後もこの作業の継続を期待したい。筆者もオブジェクト指向用語に関して同様な提案をしたが²⁾、積極的な反響はなかった。今後もあきらめずに日本語化を主張し続けていくつもりである。

コンピュータにおける英語の支配

さて、コンピュータ自身を利用する世界ではどうだろうか。ENIACが1946年にコンピュータ1号機として作られてから長い間、コンピュータは高価であり、いかに情報を小さな容量で表現するかが重要だった。今は1バイトが8ビットは当たり前だが、1960年代途中までは1バイトが6ビットという時代だった。6ビットでは高々64個の文字しか表現できない。すなわち、英字の大文字26個、数字10個、さらに10個を超える記号を表現するのが精一杯で英語の小文字ですら表現できなかった。英小文字は、1バイトが8ビットになってやっと使われるようになった。

この1960年代にコンピュータの基本的なことは完成しており、結果として英字と数字でコンピュータを使うのが基本となってしまった。長い間主流だったFORTRAN、COBOL、PL/Iなどのプログラム言語は英大文字と数字でプログラムを記述した（図1の左側）。英小文字ですら、1970年代のC言語を搭載したミニコンピュータの出現以降に研究所や大学で

やっと本格的に使われるようになった。

1980年代には漢字などのアジアの文字がコンピュータで使われるようになったが、入力装置の主流が今でも英字中心のキーボードであり、漢字入力は仮名漢字変換など間接的な手段でせざるを得ないというハンディキャップが残っている。現在のインターネットを中心としたコンピュータの世界でも英語が圧倒的に主流であり、コンピュータでは今も実質的に英語の支配が続いている。

コンピュータでの日本及び
各国の文字の利用

コンピュータで日本語を使いたいという要求は昔から根強く、1960年代から日本語利用の研究は始まっており、1970年代初めにはコンピュータを使った新聞製作など画期的な実用化も行われた。さらに、1978年の日本語ワードプロセッサの出現と1979年のJIS漢字コード(2バイト文字)の制定を契機として、メモリが比較的安くなった1980年以降に日本語利用は本格化した。

数年後には、韓国、台湾、中国などでも自国の文字コードを規格化し、アジアの各国でも自国文字のコンピュータでの利用が普及し、今では世界中のほとんどの文字が使おうと思えば使えるようになった。

標準プログラム言語での日本語の利用

さて、プログラム言語での日本語の利用はどうであろうか。この研究と開発は1960年代から始まっており、1980年代に日本語処理システムが実用段階になるに至って本格化した。最初は漢字データ処理が中心テーマであったが、ソースプログラムでも日本語を利用したいとのニーズから、現在ではCOBOL、SQL、C++、Javaなどの標準言語で名前(利用

者定義語)に日本語の指定が可能となっている(図1の右側)。

この日本語処理機能は、国際化(internationalization)機能に一般化されて国際規格化され、世界中の文字がマルチオクテット文字(複数バイト文字)として使用可能となっている。この国際規格化には筆者を含め多くの日本人が積極的に世界をリードした。この話しは筆者の思い入れの強い分野で、書きだしたら何10ページ書いても終わらないので、これ以上の言及はやめておこう。

本格的な日本語プログラム言語

1980年代後半には、さらに日本語化が進み、予約語も含めて日本語にした本格的な日本語プログラム言語が多くのメーカーで開発された。筆者もメーカー勤務時代に、予約語を日本語化した第4世代言語EAGLE/4GLを開発し、人事部や経理部など情報システム部門以外の非専門家でもプログラム作成が可能となったと好評を博した。ただし、日本語といっても、「入力データ を 作業領域 に 移す」というように、

COBOL：英語の名前を使用	COBOL：日本語の名前を使用
identification division. program-id. "gcd". environment division. data division. working-storage section. 01 big-number picture s999. 01 small-number picture s999. 01 remainder-number picture s999. linkage section. 01 positive-number-1 picture s999. 01 positive-number-2 picture s999. 01 return-number picture s999. procedure division using positive-number-1 positive-number-2. if positive-number-1 >= positive-number-2 move positive-number-1 to big-number move positive-number-2 to small-number else move positive-number-1 to small-number move positive-number-2 to big-number end-if.] perform until small-number = 0 divide big-number by small-number remainder remainder-number move small-number to big-number move remainder-number to small-number end-perform. move big-number to return-number. exit program.	identification division. program-id. "最大公約数". environment division. data division. working-storage section. 01 大 picture s999. 01 小 picture s999. 01 余り picture s999. linkage section. 01 正数1 picture s999. 01 正数2 picture s999. 01 戻り値 picture s999. procedure division using 正数1, 正数2. if 正数1 >= 正数2 move 正数1 to 大 move 正数2 to 小 else move 正数1 to 小 move 正数2 to 大 end-if.] perform until 小 = 0 divide 大 by 小 remainder 余り move 小 to 大 move 余り to 小 end-perform. move 大 to 戻り値. exit program.

図1 COBOLプログラムで最大値を求めるプログラム左側は名前に英語を使っているが、右側は名前だけを日本語化している。日本人に右側が分かり易い。しかし、予約語は英語のままであり、英語と日本語が混在しているので違和感が残る。

東京国際大学言語コミュニケーション学部 教授

日本語とプログラム言語
今城哲二

単語ごとに空白で区切る“分かち書き”という書き方を用いていた。

その後日本語プログラム言語の研究・開発は一時下火になったが、1990年代末からいくつかのチームにより本格的な日本語プログラム言語の研究・開発が進んでいる。たとえば、筆者は、“非分かち書き”の日本語プログラム言語「まほろば」³⁾を個人的に試作し、日本語化をより進化させた(図2)。インターネットで公開され、誰もが自由にダウンロードできるもので筆者が注目しているものは、教育用プログラム言語「ドリトル」(兼宗進氏開発)⁴⁾と日本語プログラム言語“なでしこ”(山本峰章氏開発)⁵⁾ ⁶⁾である。いずれも、2000年度と2004年度に経済産業省管轄の情報処理開発機構(IPA)未踏ソフトウェア創造事業のプロジェクトとして採用され、開発されたものである。

日本語プログラムの効果

筆者の経験では、COBOLで使う名前を日本語化すると、英語の名前を使っているときよりも不良数が3分の2以下に減り、テスト工数がそれに比例して少なく済んだ。さらに劇的だったのは保守要員が以前と比べ半数以下に減らせたことである。これは考えてみれば当たり前の話である。業務で使う用語は日本語なのでシステム設計段階の仕様書などでは用語は同じ日本語でそのまま記述している。それをコーディング段階以降は、使ったこともなくよく分からない英語の用語

日本語プログラム言語 「まほろば」	C言語
〔関数〕 最大公約数(正数1、正数2：整数)、 もどり値は整数。 〔名前〕 大、小、余り：整数。 〔処理〕 正数1 ≧ 正数2 のときは、 「大 ← 正数1、小 ← 正数2」 それ以外のときは、 「小 ← 正数1、大 ← 正数2」 小 = 0 になるまで、次を繰り返す。 「余り ← 除算余り(大、小)、 大 ← 小、 小 ← 余り」 もどる(大)。 〔終了〕	int gcd(int x, int y); { int b, s, r; if (x >= y) { b = x; s = y; } else { s = x; b = y; } while (s != 0) { r = b% s; b = s; s = r; } return(b); }

図2 筆者が試作した日本語プログラム言語「まほろば」で、最大公約数を計算するプログラム。右側は、同じことをC言語で書いているが、名前は一文字で記号化している。日本語プログラム言語は、日本人にとって何をやっているか分かり易い。

にわざわざ翻訳して、それを使ってコーディング・デバッグ・テスト・保守をしていたということである。日本語化すると、このような無理をせずに済み、プログラムを素直に読んで、その業務がどういうことをやっているかが分かる。

残念ながら、まだ日本のほとんどで、特にC言語やJavaを用いるときはすべてといていいほど、コーディング段階以降は英語の名前を使っている。一方、アメリカではシステム設計・コーディング・デバッグ・テスト・保守の全工程で一貫して自国語の英語を使用している。これでは、日本がシステム開発・保守の生産性・信頼性で彼らに負けるのは明白である。このような実は深刻な実態を理解せずに、プログラムで使う名前は英語にするのが当たり前と思っている教員、研究者、プログラマが大きな顔をしているのが日本である。寒心に堪えない。

予約語まで日本語にしたときの私の経験は、寝転んで斜め読みでプログラムが理解できたことである。また、ある程度の論理構成力がありさえすれば、英語に不慣れな小学生・中学生・高校生やコンピュータの非専門家でも、プログラム作成ができるようになることが期待できる。

日本語によるプログラマ、数多生まれ出でよ

著書「古寺巡礼」や「風土」で有名な哲学者和辻哲郎は、昭和初期に「日本語と哲学」という論文の中で「あるということはどういうことであるか」⁷⁾について平易な日本語で論じ、日本語で哲学することの重要性を指摘した。その論文の最後に「日本語をもって思索する哲学者よ、生まれいだよ」と呼びかけている。

同様に筆者は次のように呼びかけたい。日本語でプログラムを書くプログラマ、数多生まれ出でよ。さらに、各国に母国語でプログラム書くプログラマ、数多生まれいだよ。

- 文献
- 1) <http://www.kokken.go.jp/>
 - 2) 今城哲二：オブジェクト指向用語の日本語化の提案、第45回プログラム・シンポジウム報告集、pp.189-190、情報処理学会(2004)
 - 3) 今城哲二(共著)：計算機ソフトウェア入門、朝倉書店(2000)
 - 4) <http://www.kanemune.cc.hit-u.ac.jp/dolittle/>
 - 5) <http://nadeshiko.com/>
 - 6) クジラ飛行机：日本語プログラム言語「なでしこ」、毎日コミュニケーションズ(2005)
 - 7) 和辻哲郎全集第四巻、pp.506-551、岩波書店(1962)

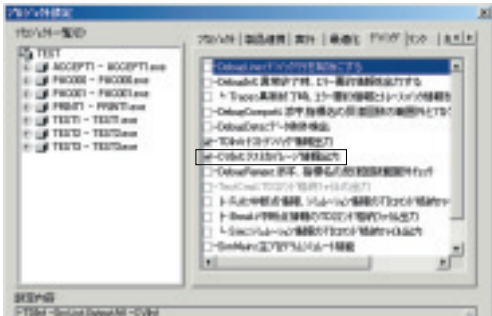
COBOL2002の便利な機能 - カウント情報

はじめに

今回はCOBOLプログラムのデバッグに有効な機能をご紹介します。テストデバッガの機能の一つとして「カウント情報」の出力機能があります。「カウント情報」とは、プログラム中の各文の実行回数のことを言います。もともとは、実行ステップが多い箇所をピックアップして実行回数を減らす工夫をするなどの性能向上対策を目的とした情報ですが、文の実行回数がわかると、「判定の誤りにより想定外の文が実行されている」「1回余分にループしている」などの簡単なロジックミスが一目で発見できることがあります。カウント情報だけで詳細な原因まで必ずしも特定できるとは限りませんが、誤りの箇所がわかるので、原因の特定がしやすいという利点があります。プログラムを実行して思った通りの結果が得られなかったとき、是非試してみてください。それでは、カウント情報の取得方法、出力例などについて説明します。

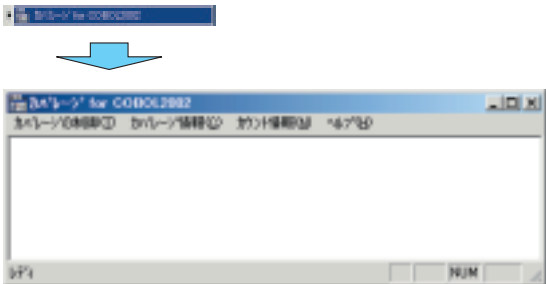
カウント情報の採取手順

手順1：コンパイラオプションの設定
カウント情報を取得するためには、コンパイラオプション「-CVInf」を設定します。開発マネージャのメニューバーから「プロジェクト」-「プロジェクトの設定」をクリックすると、「プロジェクト設定」画面が表示されます。「デバッグ」タブの「-CVInf」をクリックします。

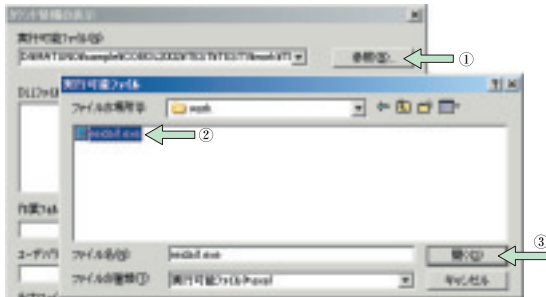


手順2：コンパイル（ビルド）
コンパイラオプションを設定したら再コンパイル（ビルド）します。

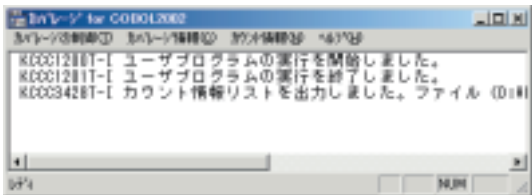
手順3：「カバレッジ for COBOL2002」を起動
「スタート」ボタン - 「プログラム」 - 「COBOL2002」 - 「カバレッジ for COBOL2002」の順にクリックします。



手順4：カウント情報の取得
「カバレッジ for COBOL2002」の画面から「カウント情報」-「表示」の順にクリックします。すると「カウント情報の表示」画面が表示されます。ここで、「参照ボタン」をクリックして実行したいプログラム名を選び、「開く」ボタンをクリックします。



手順5：実行（カウント情報の取得）
「カバレッジ for COBOL2002」に戻ったら、「OK」ボタンをクリックします。これでカウント情報を取得できました。



手順6：カウント情報の表示
カウント情報ファイル（.cnl）は、実行可能ファイルと同じフォルダに作成されます。「プログラム名.cnl」というファイルを見つけて、メモ帳等で表示してください。

カウント情報の表示例

カウント情報の表示例を示します。「実行回数」で示す数字が各文の実行回数です。行番号3100の行の実行回数が1で、行番号3300の行の実行回数が0ですから、IF文の判定は「THEN」であったことがわかります。

*****		*****	
* カウント情報 *			
*****		*****	
COBOL2002 (X) 01-02		2005-09-01 16:08:47	

プログラム名 : REIDA11			
コンパイル日時 : 2005-09-01 15:30:38			
実行日時 : 2005-09-01 16:08:47			

プログラム名 : REIDA11		プログラム名 : REIDA11	
実行回数-----		実行回数-----	
	0001700 PROCEDURE DIVISION.	1	0002900 IF 月 = 9
	0001900 Mein-Sec SECTION.		0003000 THEN
1	0002000 PERFORM 初期処理.	1	0003100 MOVE 'September!!' TO DATA2
1	0002100 PERFORM 比較処理.		0003200 ELSE
1	0002200 PERFORM 出力処理.	0	0003300 MOVE 'Not September!!' TO DATA2
1	0002300 STOP RUN.		0003400 END-IF.
	0002500 初期処理 SECTION.		0003600 出力処理 SECTION.
1	0002600 ACCEPT YYMMDD FROM DATE.	1	0003700 DISPLAY DATA0.
	0002800 比較処理 SECTION.		0003800

埼玉県立狭山経済高等学校



埼玉県立狭山経済高等学校

栗原 一也

本校の概要

本校は、茶所として知られる埼玉県狭山市の旧米軍ジョンソン基地跡地に、1985年（昭和60年）に設置された。付近の公園を「ハイドパーク」と称する点や、現航空自衛隊入間基地に隣接する関係で校舎に防音加工が施されている点などは、その名残であるといえる。

創立当時、流通経済科・会計科・情報処理科という商業に関する小学科3つから成る高校は、非常に斬新なものであった。そのため、早くも開校2年目には旧文部省から「教育課程の改善のための研究開発校」に指定されるなど、様々な意味で「モデル校」としての役割を担った。新しいタイプの学校を創ろうとする当時の気概は、次に示す「重点目標」によく現れている。

重点目標

生徒の「在り方生き方」に応じた教育の充実

- (1) 高度な資格取得の推進
- (2) 情報処理（現在は情報）教育の徹底
- (3) 英語教育の重視（現在は充実）
- (4) 大学進学への奨励（現在は推進）

細かな文言の改訂や教育課程の改編があったものの、この「重点目標」の精神と3学科構成は、現在も受け継がれている。開校21年目を迎えた本校の学校案内が、「教育目標」や「校訓」以上に「重点目標」の4本柱をアピールしている点などは、その一例である。

進路については、取得資格を活かした進学が中心

であるが、毎年約3分の1の生徒は公務員や民間企業に就職している。進路指導は、生徒の希望にそって進められ、ほぼ全員がその実現を果たしている。こうした実績は、進路指導部を中心として行われる計画的な指導に、日常の身だしなみ指導や、授業を大切にする校風等々が相俟ったものと考えられており、今後も継続していく所存である。

情報教育の徹底

すべての生徒にコンピュータを使いこなせる知識・技能を身につけさせるため、コンピュータは電子計算機実習室・情報総合実習室・実践室・ワープロ実習室・LL教室の5室に設置されている。これらの部屋には、クライアント／サーバシステムが導入されており、埼玉県公立学校間ネットワークによって外部との通信も可能である。また、各ホームルームに設置されたPCからもインターネットの利用が可能で、生徒はいつでも調べ学習等に利用することができる。

本校の特徴は、各部屋のマシンのメーカーがすべて異なるという点である。クライアントPCだけをみてもIBM・富士通・NEC・Compaqなどとなっている（一部、機種更新準備中）。このような現在の構成は単に入札の結果であるが、以前から「卒業までに色々なメーカーの製品に触らせたい」とする基本姿勢があり、マシンを管理する教員の負担よりも生徒のスキルアップを願った先達の姿勢には頭が下がるばかりである。

資格取得への取り組み

情報処理科（各学年2クラス）で取得を目指す資格は、全国商業高等学校協会の情報処理検定（プログラミング部門）1級と、経済産業省の情報処理技術者試験（基本情報・初級シスアド）である。

1年次に全商情報処理検定1級の学習を通してCOBOLをマスターし、2年次に情報処理技術者試験と表計算ソフトに関する学習を行い、3年次にコンピュータグラフィックの基礎を学ぶとともにデータベースを用いたシステム開発（マニュアル作成を含む）を行うというのが、卒業までの大まかな流れである。

こうした教育の成果は、今年度も、ワープロ部やコンピュータ部の関東・全国規模での活躍として現れた。とくにコンピュータ部は、全国高等学校IT・簿記選手権大会<IT部門新人戦>で全国11位（昨年度は5位）になるなど、我々指導する側にとっても良い刺激となっている。

情報処理科における情報関係科目

年	科目名	単位数	備 考
1	プログラミング	6	情報検定プログラミング部門1級
2	総合実践	4	情報処理技術者試験
2	情報処理	2	情報検定ビジネス情報部門2級
3	C G基礎	2	学校設定科目
3	ビジネス情報	4	データベースシステム開発
3	情報処理	選択3	V B他
3	商業技術	選択3	ワープロ他
3	課題研究	2	Webページ作成他

COBOL学習の進め方

入学直後から始まる「プログラミング（6単位）」の授業は、情報処理科におけるコンピュータ教育の入門であると同時に、高校生活の基礎を築く時間として、最も重要視されている。

時間割上は、2時間連続授業を週3日に分ける形で展開している。基本的には、2時間で教科書1テーマの内容を説明し、配布された資料を参考に生徒が各種書類を作成してきて、次の2時間でマシン実習を行う。当然、時間内にマシン実習が終わらなかった生徒は昼休みや放課後に残業をすることになるが、その際には教員が理解できるまで助言をする。この「説明→宿題→実習」という4時間1セットを繰り返して年間20本程度のプログラミングを行うと、生徒達は①家庭学習の習慣が身につく、②知らず知らずのうちにスペルや文法を覚え、③キー入力が速くなり、④集中力や自信を身につける等々の驚異的な成

長を示すようになる。このような体験量とスピードを重視した本校のCOBOL教育を称して、情報処理科担当の教員は「力業」と呼んでいる。

なお、1テーマ（1技法）の学習は、次のように進められている。

1つの技法をマスターするまでの流れ

	学習内容	備 考
説 明 (2時間)	新しい技法の説明	教科書中心
	実習問題の分析	OHPを使用
	参考資料の配付	書類作成のヒント
宿 題	スペーシングチャートの作成	検 印
	メモリマップの作成	検 印
	フローチャートの作成	検 印
	コーディングシートの作成	検 印
実 習 (2時間)	プログラムの入力	適宜助言
	コンパイル	
	デバッグ	検 印
	テストラン（結果の印刷）	
まとめ	プログラムの印刷	検 印
	実習報告書の作成	検 印
	ファイリング	随時チェック

今後について

現在、COBOL学習を中心に据えた本校の情報教育は、難しい選択を迫られている。

というのは、全商情報処理検定の改定を受けて、プログラミング部門1級の合格者が、「努力の量」ではなく単に「学力」に比例する傾向が顕著になってしまったためである。地道に努力するタイプの生徒で、全商主催の他検定では1級に複数合格を果たす生徒が不合格になったり、経済産業省の初級シスアドを取得した生徒が不合格になったりする「ねじれ現象」の多発は、生徒達の向学心を著しく損なっている。

COBOLは、アルゴリズムの学習や論理的思考能力の育成に最適なものと考えている。また、学科の目標として基本情報技術者の取得を掲げている以上、必要な言語でもある。しかし、生徒達の労が報われない現実には、COBOL学習の存続意義さえ疑問視する声が、情報処理科内部でさえ噴出している。

検定試験の種目ごとに、受験者のレベルや学習量を考慮した目標合格率（難易度）の設定がなされてしかるべきと考えるがいかがであろうか。もし現行のまま、他検定との合格率あわせのような試験が続けば、全国的なCOBOL学習離れは一層進んでしまうように思う。COBOL技術者を絶やさぬためにも、全商のご英断を心から期待している。

スクール COBOL REPORT

事例

長崎市立長崎商業高等学校



長崎市立長崎商業高等学校

杉原 利幸

本校の概要

本校は1885年長崎区立「公立商業学校」として設立された九州最古の公立商業高校である。明治、大正、昭和と校名を変え1948年学制改革により、現在の「長崎市立長崎商業高等学校」となり、今年で創立120周年を迎える伝統ある学校である。1985年、総工費62億円で完成した校舎の校地面積は、93,120㎡で施設と共に群を抜いている。卒業生は24,000名を超え、国内外さまざまな分野で活躍している。クラブ活動も活発で、体育、文化共に優秀な成績を収めている。

2004年学科改編を行い、それ迄の商業科、情報処理科、国際経済科を一つにまとめ「情報国際ビジネス科」としてスタートし今年で2年目を迎える。情報国際ビジネス科の特徴は1年次では、全員が同じ普通科目、商業科目を履修し、2年次から自分の興味、関心、適性に応じた分野の選択制をとっている。選択する分野は、「流通ビジネス」、「会計ビジネス」、「情報ビジネス」、「国際ビジネス」の4分野を設けている。情報ビジネス分野に関しては、さらに言語系、シスアド系、マルチメディア系（来年度より追加）の、三つ

のコースから選択できるようにしている。今年度までは移行年度であるため、情報ビジネス科14学級、商業科5学級、情報処理科2学級、国際経済科1学級という構成である。

実習環境の推移

昭和47年、本校に県内で最初の情報処理科が設置された。当初はFACOM230-25でパンチカードによるCOBOLの学習であった。その処理機は、教室をまるごと占有するという、とてつもない大きなもので、まさに機械そのものであった。

昭和61年、オフコン NECシステム150/88を導入し、平成7年迄、9年間NEC ITOS COBOLで言語教育を行ってきた。

平成8年、更新時、ダウンサイジングという時代の流れもあり、Windows NT 4.0、Windows 3.5の組み合わせで、クライアントサーバー型を導入し、NEC COBOL 85に変換した。当時、GO TO文に慣れていた生徒たちにとって、構造化プログラミングへの移行には相当の戸惑いがあったようである。

平成15年更新時、OSもDOSからWindows

2000, Windows XPの時代になっておりCOBOLの選択に際しては長時間の論議を要したが、結果はスクールCOBOLに落ち着いた。平成13年からのパソコン室整備計画にともない、パソコン室も整備され現在6教室（LL室含む）クライアント216台を2台のサーバーで集中管理している。主要設備は以下の通りである。

サーバー

NEC Express 5800/120RF（Win 2000 Server）5台	
セグメント数3（学習系2、業務系1）	
NEC Express 5800/120RC（Linux 6.2）	1台
WEB専用	
COMPAQ ML370（Win 2000 Server）	1台
業務系ファイルサーバー	

クライアント

NEC MAS/6T（Win 2000, XP PRO）	172台
NEC PC-9821（Win95）	44台
（17年度更新予定）	

スクールCOBOLを使用して

昨年から使用して感じていることは、まず開発マネージャーで、どうしてプロジェクトを作成しなければいけないのかという点である。（これは以前のCOBOLでは、こんなことやってなかったのに！という単純な疑問から）システム全体の構造からと思うが、その理由は今もよく理解できていない。次に、新規プログラムを作成する毎に、プロジェクトを作成させていた為、作成に要する手間暇が気になりはじめた。これに関しては、プロジェクトは最初1回だけ作成しておき、プログラムが完成後、ソースリストと実行ファイルのみを別のフォルダーに保存させ、新規ソースリストは同じプロジェクト上で同じファイル名で作成させる（上書き）方法を取っている。エディタに関しては、各領域が区別表示されるため入力し易く、予約語も色別表示される為生徒には好評である。又、論理エラーに関

してもこれ迄は個別指導で悪戦苦闘していたが、デバッグ操作が簡単にできるため、これを覚えさせることで生徒自ら解決するようになってきている。わずか1年足らずであるが、以前のCOBOLと比較して特に感じていることは、以上のような事である。

学科改編後の課題

本校は情報処理科設置以来、一貫してCOBOLによる言語教育を行ってきた。学科改編後も選択制の中で2年次よりその教育は行っているが、実施年度の今年度、言語系を希望した生徒数はシスアド系のそれと比較すると、わずか30分の1という非常に少ないものであった。来年度よりマルチメディア系が選択肢として追加される事を考えれば希望者は更に減少するのではないかと懸念される。次に、言語学習で培われる論理的な思考能力を他のシスアド系、マルチメディア系の学習の中でいかにして繁栄していくかという点、又、マルチメディア系にあっては資格取得、就職との絡みも考えなければならないという課題が残っている。

企業におけるCOBOL活用

医薬品卸業界における 情報システムのオープン化



はじめに

経済のグローバル化や市場ニーズの多様化に伴い、経営環境が激変している今日、企業間競争に勝ち残るためには、常に新しいビジネス戦略を展開するとともに、経営コストの適正化を図ることが求められています。このような状況下で、近年、メインフレームを中心としたレガシーシステムを活用してのオープンシステムとの連携や、基幹システム自体をオープン環境へ移行するなどの動きが活発化しています。

オープン化にも2種類あって、メインフレームを残して一部の業務をオープンシステムに移行する形態と、メインフレームを完全になくしてオープンシステム化する形態があります。一般的には、前者のようにメインフレームを縮小する形態が多く見られます。今回は、このオープン化に関する事例を紹介します。

情報システムのオープン化

医薬品卸業界では、商品の提供と合わせて医薬品に関する情報提供が重要になってきています。本事例の場合、現行メインフレームでは、システムの維持・運用コストがかかるだけでなく、環境の変化に柔軟な対応が、できないという問題をかかえていました。今回、取引先からのEOS（電子発注システム）やインターネットを利用したEDI（電子データ交換）の要望にも積極的に応えていくため、メインフレーム上で稼働している基幹システムを、Webやモバイル環境にも対応できるオープンシステムへの構築に取り組みました。

日立COBOLの採用

メインフレームの基幹システムは、COBOLで自社開発したものであり、長年に渡り稼働しつづけ、ノウハウが蓄積されブラッシュアップし続けた業務ロジックは、複雑な処理となっています。この業務ロジックをCOBOL以外の言語へ移行することは、非常に困難で、コストや時間がかかるため、このCOBOL資産を有効活用することが、求められました。

また、社内には、優れたCOBOLの技術者を数多く保有しています。加えて、COBOL2002規格が制定されCOBOLの将来性に不安がなくなったこと、COBOLの保守性が高いことから、プログラミング言語としてCOBOLを選択し、日立COBOL2002を採用しました。

日立オープンミドルウェアを活用した Web・モバイル環境へスムーズに移行

COBOL業務ロジックを有効活用しながら、Windows、Webやモバイルコンピューティングなどの環境にスムーズに移行することが求められました。そこで注目したのが、基幹業務システム基盤として、数多く企業に採用されている分散トランザクションマネージャ「OpenTP1」と、そのWeb化を短期間で実現する「uCosminexus/OpenTP1 Webフロントエンドセット」です。この製品を使用することによって、ExcelやVisio上で、画面入出力チェックや複雑な画面遷移のJava™やXML、JSP、JavaScriptプログラムをコーディングレスで作成し、作成されたJava™からCOBOL業務ロジックを呼び出すことができるので、Webシステムと連携する基幹システムを効率よく開発できます。

そして今回は、ソフトウェア設計モデルの一つであるMVCモデルに則って分散開発も行いました。MVCモデルとは、処理の中核を担う業務ロジック「Model」、表示や出力を司る画面「View」、画面遷移を制御する「Controller」の3階層でシステムを組み込むことです。

その結果、業務ロジック部分「Model」には、既存のCOBOL資産を有効活用して使用することで、デバッグやテスト作業の簡略化にもつながりました。新規開発する画面「View」や画面遷移「Controller」などは、ExcelやVisioに組み込まれた「uCosminexus Developer」の画面作成機能を使用することによって、作業を分担し平行して開発できるため、短期間で効率のよい開発を行うことができました。

Webシステムと連携した受注・売上入力

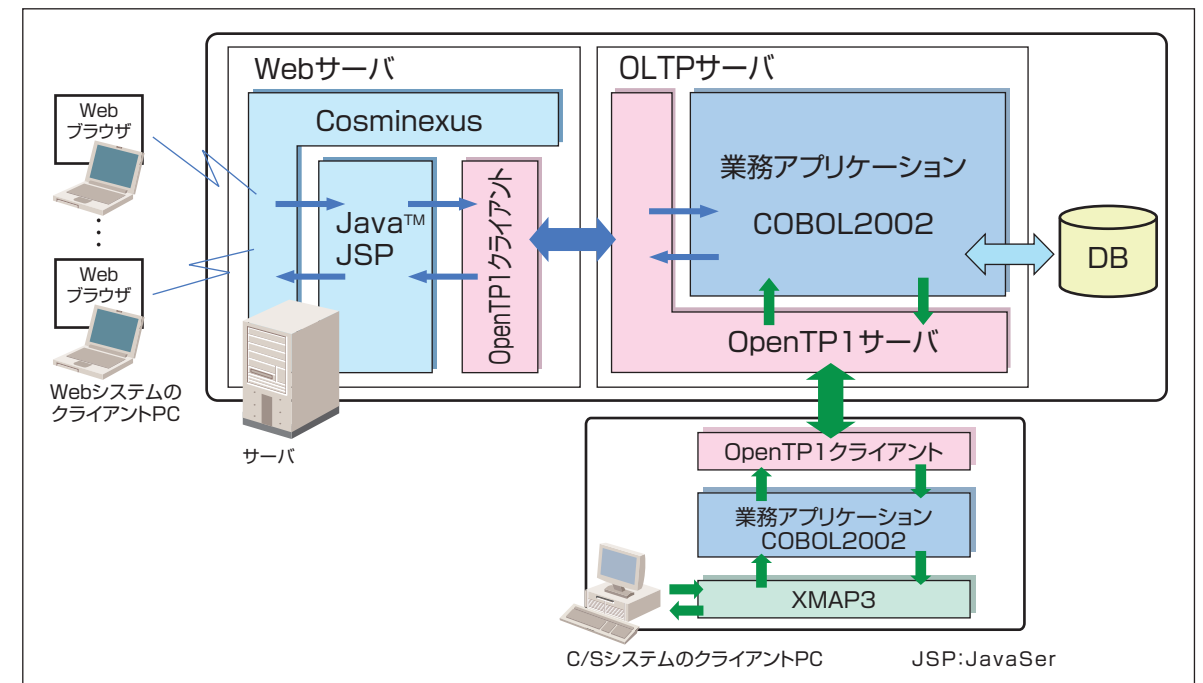
XMAP3の画面インターフェースで作成した、受注・売上入力業務を行うクライアント/サーバ型のCOBOL業務プログラムは、Webサーバから経由して基幹システムのCOBOL業務プログラムにアクセスすることが可能のため、OLTPサーバ上で稼働する同一のCOBOL業務プログラムを共有でき、基幹システムとWebシステムがシームレスに連動しました。

これにより、受注・売上入力がWebブラウザからできるようになりました。取引先とのシステム連携も作り込みやすくなり、顧客サービスを高めるために新たな戦略を展開する可能性が開けてきました。Web対応により、新製品の紹介等、鮮度の高い情報を短期間で効率的に提供できるようになりました。

基幹システムの維持・運用コストも大幅に低減でき、COBOL資産の活用で移行費用も低くおさえることができました。更に通常のオンライン業務のレスポンスも向上し、利用者からも喜ばれています。

成果と今後について

今回の移行作業でCOBOLプログラムの修正は、主に4つのインタフェースの修正がほとんどを占め、中核となるビジネスロジック部分の修正はほとんどありませんでした。4つのインタフェースとは、オンライン、画面、帳票、DBアクセスです。これらのインタフェース部分を部品化、局所化しておくことで移行性がよくなるということもわかりました。既存のCOBOLシステムを活かしつつ、「将来の移行」をスムーズに行うための基盤も整いました。「医薬品」の卸業から「情報」の卸業へと変身し、更なる発展を目指しています。



サブプログラム構成のプロジェクトファイルの作成手順

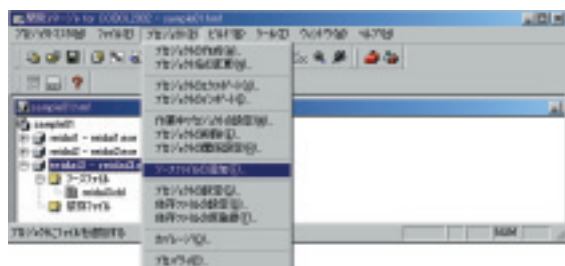
1つのプログラムが複数のサブプログラムから構成される場合のプロジェクトファイルの設定方法について説明します。
(reidai3のプロジェクトファイルに追加します。)

- ・メインプログラム:「reidai3.cbl」
- ・サブプログラム1:「reidai3a.cbl」(初期処理)
- ・サブプログラム2:「reidai3b.cbl」(比較処理)
- ・サブプログラム3:「reidai3c.cbl」(出力処理)

手順1

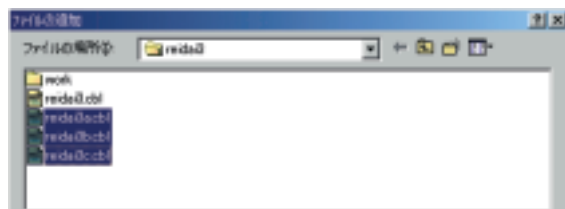
メインプログラムと同じフォルダにサブプログラム1/2/3を作成します。

- ・開発マネージャのメニューバーの「プロジェクト(P)」をクリックし、プルダウンメニューの中の「ソースファイルの追加(F)」をクリックします。
- ・ファイルの追加画面が表示されます。



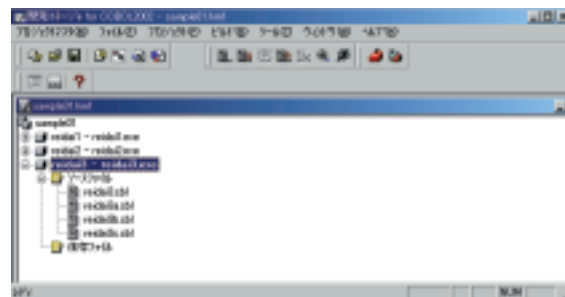
手順2

追加するファイルを選択し、開くボタンをクリックします。(複数選択可)。



手順3

「reidai3」のプロジェクトファイルにサブプログラムとして3つのファイルが追加されます。
この後、通常のコンパイルを行い、エラーがなければ実行できます。



コメント

- ①メインプログラムの終了は「STOP RUN」、サブプログラムの終了は「EXIT PROGRAM」としてください。
- ②メイン側のプログラムからサブ側のプログラムを呼び出す場合のプログラムID指定は必ずアポストロフィで囲んでください。
[例] CALL 'REIDAI3A' USING DATA0 YYMMDD.
尚、プログラムIDは大文字を使用してください。(*1)
- ③プログラム間で使用する引数のエリアはVALUE定義はできません。
各プログラムの手続き部で値の設定を行ってください。

(*1) 大文字を推奨する理由

通常のプログラミングでは大文字、小文字のどちらでもご使用になれます。

しかし、プログラムIDに小文字を使用した場合、' ' (アポストロフィ) で囲まないとコンパイラは大文字と解釈してしまい、外部参照でエラーとなってしまいますケースがあります。

[例]

- ・呼ばれる側のプログラム→PROGRAM-ID. reidai3a.
 - ・呼ぶ側のプログラム→CALL 'reidai3a' USING DATA0 YYMMDD.
- のとき外部参照でエラーとなる。
(reidai3a→REIDAI3A, 'reidai3a' →reidai3aと解釈されるため)

編集後記

本冊子の発行も4年目となります。毎回、本当にお役にになっているのか? と思いながら発行しています。今年の4月から9月までに、新しく「スクールCOBOL 2002」を採用していただいた学校が6校増えました。少しは、本冊子がお役に立てたのかと思っています。新しく採用して下さった学校の皆様有難うございます。

本冊子のバックナンバーは下記のHPに掲載されています。ご覧ください。

(<http://www.jikkkyo.co.jp/index.html>)

現在、本冊子の発行は、年2回発行していますが、今後は年1回(11月予定)の発行となります。今後もより良いレポートの発行を目指して頑張りたいと思います。ご期待ください。



発行元: 実教出版株式会社 東京都千代田区五番町5 Tel. (03) 3238-7777 <http://www.jikkkyo.co.jp/>
株式会社 日立製作所 ソフトウェア事業部 販売推進部 東京都品川区南大井6-26-2 大森ベルポートA館
Tel. (03) 5471-2339 <http://www.hitachi.co.jp/soft/>

※Java 及びすべてのJava関連の商標及びロゴは、米国及びその他の国における米国Sun Microsystems, Inc.の商標または登録商標です。
※Microsoftは、米国およびその他の国における米国Microsoft Corp.の登録商標です。Microsoft Excelは、米国Microsoft Corp.の商品名称です。
※Visioは、米国およびその他の国における米国Microsoft Corporationの登録商標です。
※Windowsは、米国およびその他の国における米国Microsoft Corp.の登録商標です。
※その他記載の会社名・製品名は、それぞれの会社の商標もしくは登録商標です。

日立スクールCOBOL85
リピータ 特別価格のご案内

既に「日立スクールCOBOL85スクールバック」又は、「スクールCOBOL85 GUI構築スクールバック」をご導入されている場合、最新の「日立スクールCOBOL 2002スクールバック」「スクールCOBOL2002GUI構築スクールバック」をリピータ特別価格でご提供させていただきます。