スクールCOBOLユーザーの方のための情報誌

スクール COBOL Report

Contents

●巻頭寄稿
プログラミング「体験」教育の必要性
筑波大学ビジネス科学研究科 教授
久野　靖

●「COBOL 最新企画」
Visual Basic .NETからCOBOLアプリケーションを呼び出す方法のご紹介

●ユーザー校事例
千葉経済大学附属高等学校
清水 隆治
埼玉県立皆野高等学校
木村 敏規

●スクールCOBOL2002の便利な機能

Vol.10
2007. 3

情報教育をサポートするじっきょうのテキスト

◇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
2007年度版 **速攻基本情報技術者 午前**
A5判 200p. 定価 2,205円

合格のための総仕上げ
2007年度版 **速攻基本情報技術者 午後**
A5判 192p. 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◇

2007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
B5判 280p. 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3-2-1 日本生命博多駅前ビル
TEL 092(473)1841 FAX 092(471)7529

プログラミング「体験」教育の必要性

筑波大学ビジネス科学研究科 教授
久野 靖 *Yasushi KUNO*

1984年　東京工業大学理工学研究科単位取得退学。
同年　　東京工業大学理学部情報科学科助手。
1989年　筑波大学経営・政策科学研究科（現ビジネス科学研究科）経営システム科学専攻講師。助教授を経て現在、教授。
理学博士。
プログラミング言語/言語処理系、ユーザインタフェース、プログラミング教育、情報教育などに興味を持つ。
著書：「UNIXの基礎概念」「入門JavaScript」「情報科教育法（共著）」ほか。
情報処理学会、日本ソフトウェア科学会、ACM、IEEE-CS各会員。

後れている日本の情報教育・情報処理教育

筆者が所属している情報処理学会・情報処理教育委員会では、2005年秋から昨年にかけて、いくつかの提言を出している。そのトップバッターとなったのが「日本の情報教育・情報処理教育に関する提言2005」であり、現在はこれにパブリックコメントを頂いた結果、追補・改訂を行った版を公開している[3]。

そこに何が書かれているかというと、ひとことで言えば「日本では情報社会に向けたインフラ整備はやっているが、人材育成の面では非常に立ち後れている」「それを何とかするために、情報教育・情報処理教育の改革が必要だ」ということである。

ここまで読まれて意外に思われた方もいるかも知れない。「日本は技術立国を目指していて、情報技術にも多く力を入れているはずではなかったのか?」「世の中の人は皆パソコンや携帯端末を使いこなしているではないか?」と。

確かに、国は学校におけるPCやネットワーク接続といった設備の充実にはそれなりに力を入れている。しかし、人材の育成、つまり有効な教育という観点でいえば、設備を入れただけでは何もできたことにならない。もちろん、何も習わなくても誰でも携帯くらい使えるだろうけれど、それこそ「自動車の運転ができるからといって、自動車の製造技術が身についてるわけでは全くない」のである。

ここで「情報社会に向けた人材育成が難しいのはどこの国でも同じでは?」と思われるかも知れない。しかし本当にそうだろうか。OECDの国際学習到達度調査（PISA）で日本の生徒の学力がやや低下したのが問題になったが、同時に実施された情報機器の活用能力に関する調査[1]はもっとショッキングであり、15歳の生徒がコンピュータを利用する割合でいうと日本はOECD加盟国中で最下位に近いレベルである。さらに、「コンピュータをプログラミングのためによく使う」という割合は、米国で33%、OCED平均で23%だが、日本はわずか3%とそれこそ最下位である。

2006年秋の「中等教育における情報教育国際会議（ISSEP2006）」で、一橋大学の兼宗先生がこれらの状況を報告したところ、各国の参加者から「このままで日本は大丈夫なのか?」と本気で心配する意見を頂いたほどである。[2] 本当にこれでいいのだろうか?

世の中でどんな問題が起きているのか?

情報・情報技術に関する教育の立ち後れを指摘すると「情報技術は単なる道具なのだから、別にその内容を理解させる必要などない」という反論を頂くことがある。しかし本当にそうだろうか?

上記の提言を公開した後、2005年末から2006年初等に掛けて、情報や情報技術に関する無知・無理解が影響を及ぼしていると思われる事件が立て続けに起きた。たとえば「耐震偽装事件」では、提出した計算書が「大臣認定プログラムが出力したらしく見えていれば」よいことを利用して偽の計算書を通していたし、「東証1円61万株事件」では、「値が通常範囲を超えているという警告はいつも当り前に出ていてそれを無視して注文を出すのが通常」だったことが問題の要因の1つになっていた。情報処理教育委員会では、その他の問題も含め、改めて世の中に警告すべく追加のコメント[2]を出している。

これらで結局何が問題かというと、情報や情報技術についてある程度分かっているなら誰かが「それは何とかしないと問題でしょ」と言うはずのことがらがずっとそのままになっていて、その結果大トラブルが発生している、という点である。

これらの問題を未然に防ぐには、これらのシステムや制度に関わる誰もが情報や情報技術について一定水準の理解を持っていて、その中から「問題だよね」という気付きが生まれることが必要である。

しかし現在のわが国では、現場の人からトップに至るまで誰もが「コンピュータのことは自分は分からないから専門家に任せておけばよい、自分はもっと大事な上位のことがらを考えるのが仕事だから」と言い訳して情報技術者に丸投げしている状態である。それでは技術者が間違った判断や設計を採用してもそれを止めるものがない（そもそも情報技術者は顧客業務の専門家ではないので、その状態で丸投げすることは本来無理がありすぎる）、そこに様々な問題の根っこがある、というのが我々の考えである。

情報技術者教育にも問題が…

上のような問題の一方で、情報技術者の育成自体にも問題がある。たとえば経団連は現在のわが国のICT（情報・通信技術）人材が質・量ともに不足していることを問題とし、その育成強化を提言している[5]。

我々は、ここでの大きな問題は（1）ICT関係の専門家を志望する生徒・学生が少ないこと、（2）間違った理解をもとにICT関係の専攻を選択し、早々にドロップアウトする生徒・学生が多数いること、であると考えている。実際、情報専門学科の学生に「あなたはなぜこの学科に来たの?」と質問したところ、「オフィスソフトを上手に使えるようになりたかったから」との答えを得たという実話がある。

ではどうしたらいい?

では、前節までで述べたような問題を克服するにはどうしたらいいのだろう? それには地道に、情報教育・情報技術教育を改革していくしかない、ということになり、そこで冒頭の「提言」が出て来たわけである。そこでは、次のことがらの実現を求めている。

プログラミング「体験」教育の必要性

筑波大学ビジネス科学研究科 教授
久野　靖

（1）小学校・中学校・高等学校それぞれの発達段階に応じて適切な「手順的な自動処理」の体験を持たせる。
（2）高等学校の教科「情報」に選択科目を追加することで「手順的な 自動処理」に関心を持った生徒が系統的に学べる場を設ける。
（3）「情報」を学び、得意とする生徒が多様な分野へ進学できるように、大学の入試に「情報」に関する内容を追加する。
（4）大学の一般情報教育において、「手順的な自動処理」についての制作体験をさせる。また各専門において、その専門に関連した情報系科目を選択可能とする（教員養成系においては必修とする）。
（5）大学の情報関連学科において、「手順的な自動処理」の構築に対する適性を持つ学生を対象とし、将来高度ICT人材として活躍できる水準の教育を行う。
（6）大学・大学院の情報分野を含む各専攻において、「手順的な自動処理」の構築に対する適性を持つ社会人を学生として受け入れ、各分野におけるICT人材として活躍できる水準の教育を行う。
これには（1）～（5）の教育を行うことができる教員の養成および研修を含む。

「手順的な自動処理」という言葉が目につくが、これはコンピュータが予め決められた手順を自動的に実行していくことで動作している、という原理を表している。それならソフトウェアとかプログラムと言えばいいのでは、と思われるかも知れないが、情報教育方面では「プログラミング」という言葉が非常に大きな反発を招く傾向があることと（これ自体困ったことではある）、表計算ソフトの自動計算機能、音楽ソフトの自動演奏なども含めた広い概念ということでこの用語を採用している。

我々が求めていることは、小学校・中学校などの早い段階から、これらの「自動的に～する」ものを「体験」し、高校まで来たら簡単なプログラム様のものを「体験」する、その「体験」は（数時間程度でもいいので）全員に持って欲しい、ということである（ことさらに体験を強調しているのは、従来型の「言語の構文や機能を逐一学習する」教育はこの段階では求めないためである。しかしその段階でも手順である以上、「自ら考えて作る」ことは必須である）。

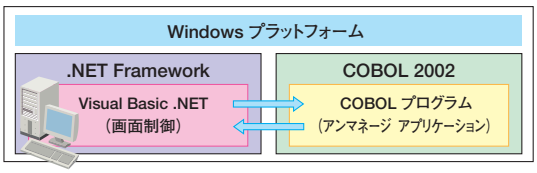
そのような体験に根ざして知っていれば、コンピュータにはどんなことができてどんなことができないかをある程度まで正しく判断できるようになり、上述の「おかしなこと」に疑問を出せるような、国民全体の「情報水準の底上げ」が実現できるのでは、と期待している。そしてもう1つ、これらの体験から自分の適性を見出した生徒が、専門高校、選択科目、情報専門学科などに進むことによって、将来のわが国を担うICT人材となって欲しい、というのも大いなる期待である。

文献
1）OECD, Are Students Ready for a Technology-Rich World?: What PISA Studies Tell Us.
http://www.pisa.oecd.org/document/31/0,2340,en_32252351_32236173_35995743_1_1_1_1,00.html
2）兼宗　進、日本の情報教育と可能性。
http://blog.nikkeibp.co.jp/pconline/edu/2006/12/kanemune30.html
3）情報処理学会情報処理教育委員会、日本の情報教育・情報処理教育に関する提言2005（2006.11改訂/追補版）。
http://www.ipsj.or.jp/12kyoiku/teigen/v81teigen-rev1a.html
4）情報処理学会情報処理教育委員会、2005年後半から2006年初頭にかけての事件と情報教育の関連に関するコメント。
http://www.ipsj.or.jp/12kyoiku/statement2006.html
5）日本経済団体連合会、産学官連携による高度な情報通信人材の育成強化に向けて、2005。
http://www.keidanren.or.jp/japanese/policy/2005/039/index.html

Visual Basic .NETからCOBOLアプリケーションを呼び出す方法のご紹介

はじめに

今回はMicrosoft社のアプリケーション動作環境である「.NET Framework」上のアプリケーションからWindows上のCOBOLアプリケーションを呼び出す方法をご紹介します。ご存知のように.NET Framework は.NET対応のアプリケーションが動作する環境です。.NET Framework上のプログラミング言語には、Visual Basic .NET（Microsoft Visual Basicの次世代言語）、C#、J#などがあります。今回紹介するのは、Visual Basic .NETで作成したアプリケーションから、通常のWindows版COBOL2002で作成したアプリケーションを呼び出す方法です。もちろん、スクールCOBOL2002で作成したCOBOLアプリケーションも呼び出せます。この、Windows上のCOBOLアプリケーションのことをアンマネージアプリケーションといいます。呼び出しの概念図を次に示します。



Visual Basic .NETのユーザインターフェースは、「スクールCOBOL2002 GUI構築パック」と同様に、「フォーム」という基本ウインドウにボタンやスクロールバーなどの部品を貼り付け、プロパティを設定するなどして作成します。そして、イベントに対する処理手続きをプログラミングして完成させます。

Visual Basic .NETからCOBOLアプリケーションを呼び出す方法

Visual Basic .NETのプロシージャ定義や関数定義から、ダイナミックリンクライブラリ（DLL）として作成されたCOBOLアプリケーションのプログラム定義を呼び出すことができます。

以下に、「COBOLのデータ型とVisual Basic .NETの宣言」および「Visual Basic .NETおよびCOBOLプログラム作成の方法」を説明します。なお、以下の説明は、最後に示す例題プログラムの説明にもなっていますので、例題プログラムと照らし合わせながら読むと理解し易いでしょう。

■COBOLのデータ型とVisual Basic .NETの宣言

①Visual Basic .NETとCOBOLのデータ型の対応表を次に示します。

データ型	Visual Basic .NET（送り出し側作用対象）	COBOL（受け取り側作用対象）
バイト型	Dim a As Byte	01 a PIC 9（1）COMP-X
短整数型	Dim a As Short	01 a PIC S9（4）COMP
整 数 型	Dim a As Integer	01 a PIC S9（9）COMP
長整数型	Dim a As Long	01 a PIC S9（18）COMP
文字列型	Dim a As String	01 a PIC X（n）

凡例）n:文字列の長さを示す。

②引数を渡す方法

引数を渡す方法は、値渡し（ByVal）と参照渡し（ByRef）があります。

値渡しを使用した場合、COBOLプログラムで値を変更しても、Visual Basic .NETのデータ項目の内容には影響を与えません。

参照渡しを使用した場合、COBOLプログラムで値を変更すると、Visual Basic .NETのデータ項目の内容も変更されます。ただし、文字列型は、参照渡し（ByRef）を使用できないため、次の組合せとなります。

「Visual Basic .NET値渡し（ByVal）→ COBOL参照渡し（BY REFERENCE）」

■Visual Basic .NETプログラム

①Visual Basic .NETプログラムから呼び出す、COBOLプログラムのDLLの関数宣言をします。

エイリアス名は、「_プログラム名@n」（注1）になります。

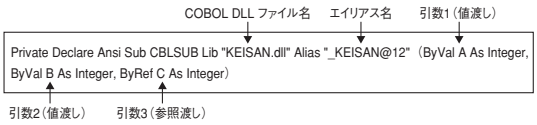
例）下記のCOBOLプログラムを呼び出す関数宣言

COBOL DLLファイル名：KEISAN.dll

プログラム名：KEISAN

引数：整数型の値渡し（ByVal） →２個

整数型の参照渡し（ByRef） →１個



注1：nの計算式は、次のとおりです。

n＝(引数の数×4)＋(値渡しで受け取る倍精度浮動小数点数字項目引数の数×4)

②Visual Basic .NETプログラムからGUIモードのCOBOLプログラムを呼ぶ場合、COBOLプログラムを呼び出す前後で「CBLGINTサブルーチン」と「CBLENDサブルーチン」を呼び出す必要があります。つまり、次の順番に呼び出します。

・「CBLGINTサブルーチン」の呼び出し

・COBOLプログラム（DLL）の呼び出し

・「CBLENDサブルーチン」の呼び出し

サブルーチン関数の宣言は次のようになります。

例）CBLGINTサブルーチンの宣言

Private Declare Ansi Sub CBLGINT Lib "CBL85RT.dll" Alias "_CBLGINT@0" （）

例）CBLENDサブルーチンの宣言

Private Declare Ansi Sub CBLEND Lib "CBL85RT.dll" Alias "_CBLEND@0" （）

③CALL文で、①のCOBOL ダイナミックリンクライブラリ（DLL）や、②のサブルーチンを呼び出します。

例）①のCOBOL ダイナミックリンクライブラリ（DLL）の呼び出し

Call CBLSUB（DATA1, DATA2, DATA3）

例）②のCBLGINTサブルーチンの呼び出し

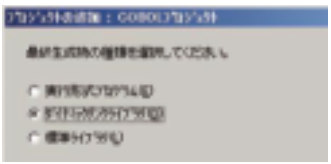
Call CBLGINT（）

例）③のCBLENDサブルーチンの呼び出し

Call CBLEND（）

■COBOLプログラム

①COBOLプログラムは、stdcall呼び出し規約で、ダイナミックリンクライブラリ（DLL）として作成します。
ダイナミックリンクライブラリは、COBOL2002の開発マネージャを使用して、「プロジェクトの作成」で次の設定をすることにより作成できます。
＜「最終生成物の種類」として「ダイナミックリンクライブラリ」を選択＞



＜「プロジェクトの種類」として「-DII指定（DLLを作成する）」を選択＞

＜「DLL呼び出し規約選択」画面で「DLLの属性をStdCallにする」を選択＞

②連絡節に、Visual Basic .NETプログラムから渡ってくる引数の数だけ、データ項目を定義します。
例）「整数型の値渡し → 2個、整数型の参照渡し → 1個」の引数が渡ってくる場合

```
LINKAGE SECTION.
  01 DATA1 PIC S9（9）COMP.
  01 DATA2 PIC S9（9）COMP.
  01 DATA3 PIC S9（9）COMP.
```

③②で定義したデータ項目を受け取るため、手続き部の見出しにUSING指定をします。引数の渡し方には、値渡し（BY VALUE）と参照渡し（BY REFERENCE）があります。
例）「整数型の値渡し → 2個、整数型の参照渡し → 1個」の引数を受け取る場合

```
PROCEDURE DIVISION USING BY VALUE          DATA1
                         BY VALUE          DATA2
                         BY REFERENCE      DATA3.
```

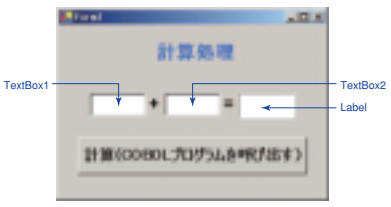
例題プログラム

Visual Basic .NETのフォームに数値が入力され、計算ボタンが押されたら、COBOLプログラムを呼び出して計算する例題プログラムをご紹介します。

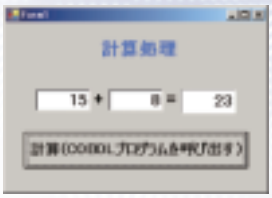
（1）Visual Basic .NETプログラムの作成

①Visual Basic .NETの画面を作成

Visual Basic .NETのフォームに、テキストボックスやラベル、ボタン等の部品を貼り付けて、次の様な画面を作成します。Visual Basic .NET部品の名前を矢印で示します。入力されたデータをVisual Basic .NETプログラムで参照するときは、この部品の名前を使用します。



実行結果の画面例を次に示します。「＋」の左右のテキストボックスに加算する値を入力して「計算」ボタンを押すと、COBOLプログラムが呼び出されて計算を行い、「＝」の右側のラベルに結果が表示される仕様です。



②COBOLプログラム DLLの関数宣言やサブルーチン関数宣言の作成

```
'COBOLプログラム（KEISAN.dll）を呼び出す関数宣言
Private Declare Ansi Sub CBLSUB Lib "KEISAN.dll" Alias "_KEISAN@12"
（ByVal A As Integer, ByVal B As Integer, ByRef C As Integer）

'COBOLプログラムをGUIモードで実行するためのCBLGINTサービスルーチン宣言
Private Declare Ansi Sub CBLGINT Lib "CBL85RT.dll" Alias "_CBLGINT@0" ()

'COBOLプログラムをGUIモードで実行するためのCBLENDサービスルーチン宣言
Private Declare Ansi Sub CBLEND Lib "CBL85RT.dll" Alias "_CBLEND@0" ()
```

③計算ボタンのクリックイベント処理の作成

計算ボタンが押されたら、入力された値を引数として、COBOLプログラム "KEISAN.dll"を呼び出す処理を記述します。

```
Private Sub Button1_Click（ByVal sender As System.Object, ByVal e As System.EventArgs）
Handles Button1.Click
  Dim DATA1 As Integer
  Dim DATA2 As Integer
  Dim DATA3 As Integer

  'CBLGINTサブルーチンを呼び出す
  Call CBLGINT（）

  '入力されたテキストボックスの値を取得する
  DATA1 = Val（TextBox1.Text）
  DATA2 = Val（TextBox2.Text）

  'COBOLプログラムを呼び出す
  Call CBLSUB（DATA1, DATA2, DATA3）

  'COBOLプログラムで計算した結果を、ラベルに設定する
  Label.Text = DATA3

  'CBLENDサブルーチンを呼び出す
  Call CBLEND（）

End Sub
```

（2）COBOLプログラムの作成

Visual Basic .NETから渡ってきた引数を受け取り、それらのデータを加算する処理を記述します。COBOLプログラムは、stdcall呼び出し規約で、ダイナミックリンクライブラリ（DLL）として作成します。DLLの名称は、"KEISAN.dll"とします。

```
IDENTIFICATION DIVISION.
  PROGRAM-ID. KEISAN.
  ENVIRONMENT DIVISION.
  DATA DIVISION.
  WORKING-STORAGE SECTION.

LINKAGE SECTION.
  01 DATA1 PIC S9（9）COMP.
  01 DATA2 PIC S9（9）COMP.
  01 DATA3 PIC S9（9）COMP.

PROCEDURE DIVISION USING        BY VALUE
DATA1
                        BY VALUE    DATA2
                        BY REFERENCE      DATA3.
*> Visual Basic .NETから渡ってきた値を元に計算する
   COMPUTE DATA3 = DATA1 + DATA2.

END PROGRAM KEISAN.
```

スクール COBOL REPORT

ユーザー校 事例

千葉経済大学附属高等学校

千葉経済大学附属高等学校
清水 隆治

本校の概要

本校は1933（昭和8）年、学園の母胎となる千葉女子商業学校として創設された。昭和23年、学制改革により千葉女子経済高等学校（商業科）となる。昭和29年、千葉経済高等学校と改称し男子部を併設した。昭和36年に建築科、昭和50年に普通科、昭和60年に情報処理科を新設し、4科を持つ男女共学の学校となる。また、幼稚園・高校・短大・四大を持つ学園でもある。平成15年には創立70周年記念事業の一環として、すべての生徒が学べる新しい学習環境を整えるため、9階建ての新校舎（最上階は展望学食）が完成した。また、制服も十数年ぶりに刷新した。建学の精神「片手に論語、片手に算盤」のもと、校風「明朗・真摯・友愛」を実践している。

現在、約1,400名強の生徒が在籍し、同窓生は4万人を超えている。しかし少子化で15歳人口が減ってきて、県内私学も生徒確保が厳しくなっていることや多様化する生徒のニーズに対応する体制作りの一つとして普通科に特進クラスを作り、難関大学を目指す生徒のニーズに対応した。また、専門的な資格（情報処理、簿記、建築）を取得したい生徒へのニーズにも答えてきている。

近年の進路状況は約95%以上が進学を希望している。また、部活動も盛んで今年は野球部が秋季関東大会優勝、神宮記念大会ベスト4が認められ、春の選抜大会初出場が決まり、夏春連続出場という快挙が最新のホットニュースである。

学習環境の推移

昭和59年 N5200モデル05　21台
　PTOSCOBOL　8インチFD
昭和62年　N5200モデル05MK II
　22台増やし一人一台に、プリンターは切誓えスイッチで二人で1台。
平成6年　N5200モデル98/80
　45台　PCゼミの導入により生徒への対応が楽になった。
平成11年　機器更新により東芝に変わる。
　クライアント・サーバ方式を採用
　サーバ　MAGNIA5000
　COBOL85
平成17年　新校舎完成に伴い情報実習室が4室となる。クライアント・サーバ方式で情報処理科用にはCOBOL2002が、建築科用にはAuto CAD、商業科用には弥生会計が、そしてNetGroupAssistによってソフトでの生徒との対応等を可能にしている。この4室のうち2室を情報処理科用として使用している。また、各室ともプロジェクターがあり、大いに活用している。

情報実習室のシステム構成

機器用途	機　種	台数	OS等
サーバ	IBM BladeServer HS20	4	WindowsServer2003 Standard Edition
	IBM BladeServer HS20	2	RedhatEnterprise Linux ES3.0
	IBM xSeries 336	1	RedhatEnterprise Linux ES3.0
	IBM xSeries 346	1	WindowsServer2003 Standard Edition
	NEC Express5800/12RF-1	1	WindowsServer2003 Standard Edition
教師用クライアント	NEC MY30V/C-G	4	WindowsXP Professional SP2
生徒用クライアント	NEC MY30V/C-G	180	WindowsXP Professional SP2
モノクロレーザプリンタ	NEC MultiWriter 1500N	23	
カラーレーザプリンタ	NEC Color MultiWriter 9250C	7	
プロッタ	EPSON PX-7000	2	
スキャナ	EPSONGT -7400U	4	
プロジェクタ	EPSON EMP-7900	4	

平成18年度の中学生を対象とした体験学習では新実習室で「スーパー実践くん」を利用して名刺作りを行った。例年以上の人数が集まり、受験生の増加を期待している。

COBOL実習の概要

教科書（最新プログラミング21）実教出版を使い、例題を説明し実習問題を進める。実習は以下の手順で行っている。

1 問題の分析、主記憶領域図、流れ図の作成
　オペレーションマニュアルに記入。点検し検印後返却
2 プログラムをコーディングシートに書く
3 点検し、ミスを訂正したのち検印し返却
4 プログラムの入力
5 コンパイルし、エラー直しをする
6 デバッグを経て、RUNする。
7 これまでの提出物と一緒に実行結果をファイルリングする。プログラムは各自のフォルダーに保存する。

日立のスクールコボル2002は、構文チェックなど様々な点で、操作性に優れているため、生徒も学習しやすいところはありがたい。

また、COBOL以外の言語は行っていない。理由としては、解決すべき課題の内容に応じて、いかに効率よく分かりやすいプログラムを作るかというアルゴリズムを考えさせるのに最適と考えためである。

情報処理科の教育課程の概略

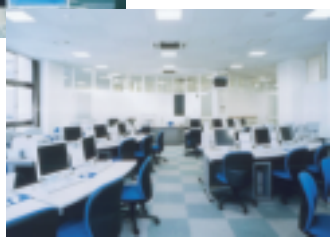
情報処理関連科目

講　座　名	単　位	学　年	備　考
ITと社会	3	1	
情報処理	3	1	
プログラミング	6	2	
情報実習	3	2	選　択
情報コミュニケーション	3	3	選　択
応用情報	2	3	ネットワーク
課題研究	3	3	関連資格取得

1年生の「情報処理」で全商情報処理検定ビジネス情報部門2級、2年生の「プログラミング」でプログラミング部門1級合格を目指している。2年生の「情報実習」ではビジネス情報部門1級を目指し、検定後はパワーポイントを使い、各自の興味あるテーマでプレゼンを学ぶ。「情報コミュニケーション」ではPhotoshop の利用やホームページビルダーでHP作成を学んでいる。

検定への取り組み

経済産業省の基本情報技術者試験及びシステムアドミニストレータ試験の取得を目指している。現在の3年生にこの2つとソフトウエア開発技術者試験の3つに合格している怪物君もいるが、現実問題として毎年多数の合格者を輩出するには至っていない。また、全商情報処理検定のプログラミング部門・ビジネス情報部門1級の合格者数も新検定が試行されて以来激減しているのは入学者のレベルの低下なのか、カリキュラム上に問題点があるのか、新検定により同一級を受験できないためなのかが気になる点でもある。



スクール COBOL REPORT

ユーザー校 事例

埼玉県立皆野高等学校

埼玉県立皆野高等学校
木村 敏規

本校の概要

本校は、昭和41年に創立された埼玉県秩父地域にあるビジネス系専門高校（商業高校）である。近隣には岩畳で知られる長瀞町や夜祭などで有名な秩父市があり、自然や文化にも恵まれた環境にある。

設置学科は商業科と情報処理科であるが、入学後の学習内容によって学科が選択できるユニット選択制というシステムを埼玉県内の商業高校で唯一取り入れており、広く生徒の興味と関心に応じた多様な科目が選択できるようにしてある。

ユニット選択制とは生徒募集を商業科・情報処理科の区別なく「商業系」として一括して募集し、1年次ではすべての生徒が共通履修を行い、その中でガイダンスにより商業系か情報処理系のどちらに興味があるかを考えさせ2年次以降の学習内容を決定し、進路希望に応じたコースを選択する。

最終学年の3年次には学科決定科目を選択することで卒業学科を決定していくというシステムである。

情報処理科教育課程

学科選択を含めたユニット選択制を取り入れていることから、情報処理科としてのプログラミング言語学習は2年次から開始することになる。情報システムユニットを選択した生徒に対してプログラミング3単位、情報処理3単位の計6単位を相互に関連させて授業を進め、9月の全商情報処理検定でプログラミング部門2級、1月の同検定で1級の取得を目標としている。

1年次から情報処理科としてプログラミングを学習させる通常のカリキュラムに比べてハードなスケジュールとなっている。

実習環境および新システムの特徴

本校では平成17年度に実施されたコンピュータシステムの更新に際し、本県の公立高校として初の

ユニット選択制度　教育課程

1年次　商業共通

ビジネス基礎	2
商業技術	2
簿記	4
情報処理	3
課題研究	1
計	12

2年次　商業共通

総合実践	2
課題研究	1
計	3

3年次　商業共通

課題研究	3
経済活動と法	3
計	6

商業共通計	21

2年次　ユニット選択必修

ビジネスカテゴリ

ビジネス会計	上級簿記	3
	原価計算	3
ビジネスリテラシー	商品と流通	3
	文書デザイン	3

インフォメーションカテゴリ

情報システム	プログラミング	3
	情報処理	3
情報リテラシー	情報処理	3
	文書デザイン	3

ユニット計	6

プラスワン

ライセンス系

会計	2

リテラシーマスター系

秘書実務	2
マーケティング	2

ライセンス系

ビジネス情報	2
文書デザイン	2

リテラシーマスター系

ネットワーク総合	2
簿記	2

進学系

国語表現	2
リーディング	2

プラスワン計	2

3年次　学科決定

商業科

ビジネス総合	4

情報処理科

情報技術総合	4

学科決定科目計	4

選択科目群A

国語表現Ⅱ	2
リーディング	2
化学Ⅰ	2
家庭看護・福祉	2
ビジネス情報	2
マーケティング	2
原価計算	2
国際ビジネス	2

選択科目群B

国語表現Ⅰ	2
地理A	2
数学基礎	2
生物Ⅰ	2
フードデザイン	2
プログラミング	2
会計実務	2
秘書実務	2

選択合計	2

「ディスクレスシステム」を採用した。(3教室、約150台に導入)

ディスクレスシステムでは各PCにはHDDは内蔵されておらず、PCの電源を入れるとサーバに登録してあるOSやアプリケーションをネットワーク経由で起動し、各PCのメモリ上に仮想HDDとして読み込む。

この方式ではサーバの処理機能とネットワーク回線の強化が必要だが、各PCのHDDを必要としないため、PCの接続台数が多いほど価格的に廉価になる。さらに、このシステムでは一時処理されたデータも各PCのメモリ上の仮想HDDに記憶され、電源を切るとすべて消滅する仕組みとなっている。仮にコンピュータウィルスが外部より進入したとしてもウィルスが潜伏したり破壊したりするHDDが存在しないため、教育機関のコンピュータシステムとして、より安全な仕組みとなっている。

言語処理プログラムの充実

近年の情報処理教育ではコンピュータの利用技術について重点がおかれる傾向にあるが、本校の情報処理科では、基本的な考え方として、コンピュータ資源を利用して解決すべき問題に合わせてデータを加工し、利用者に提供できる立場の人材育成を目標としている。

その問題解決のためのツールとしてプログラミングを学ぶことは、論理的な思考力および解決方法を創造する力を養うために最適な情報教育であると考えている。また、プログラミング言語は世界共通の技術であるため習得すれば応用の幅が格段に広がる。そのためツールである言語処理プログラムは複数用意されていることが望ましい。

本校のシステムで扱える言語の代表的なものは以下の表の通りである。

言語	開発ツールおよび統合環境
COBOL	スクールCOBOL2002 開発マネジャfor COBOL2002
Visual Basic Visual C/C++ Visual C#、J#	Visual Studio.NET2003
C/C++	Borland C/C++ BCC Developer
Java	Java2 SDK Eclipse

実習時の手順･方法

生徒にとって初めてのプログラミング作業が分かりやすくスムーズに作業が行えることは、それ以降のプログラミング学習に大きな影響を与える。そのため最初の実習については十分に注意を払って指導を行う必要がある。

本校ではCOBOL言語の習得をより分かり易くするために簡単なCOBOL言語仕様および実習の手引き等を手作りで作成し利用している。

特にCOBOL言語の特徴的なデータ部のファイル節および作業場所節はスペーシングチャートを基に独自の領域図を作成し、領域図の記述通りにコーディングしていくとデータ部が完成するように工夫をしている。

また、手続き部に関しても独自に作成したCOBOL言語プログラムイメージ図を基に、コンピュータがCOBOL言語を利用して作業するときの考え方や流れをフローチャートとコードで追いながら説明を行いコーディングしている。

2回目以降の実習は1回目のプログラムに改良を加えていく内容であるが、生徒には新たに最初から作成し直すように指導している。

スクールCOBOL利用時の工夫点

生徒がCOBOL言語で作成した自分のアプリケーションプログラムをWindows上から実行して実感してもらうために、プロジェクトの作成をする際に実行形式ファイルの作成場所に特定のフォルダ（EXEという名前のフォルダを作成）を指定し、そのフォルダに出力するように指導している。

さらに、入出力節のファイル管理記述（SELECT句）の出力装置名には前述のEXEフォルダにテキストファイルが作成されるように出力ファイルを指定している。

こうすることにより、EXEフォルダ内に
- データソースリスト
- 作成したアプリケーションプログラム
- 実行後出力されたテキストファイル

の3つのファイルをまとめ、作成したアプリケーションを単体で起動し、生徒自身がプログラムの機能と結果を実感して確かめることが出来る。

また、実行結果をテキストファイルに出力するようにしているので、開発マネジャ上で実行しても結果がすぐには分からない。そのため、WRITE文を記述した後に必ずDISPLAY文を記述するように指導している。この指導により、実行すると同時に結果が画面表示されるので非常に便利である。

最後に

COBOL言語は幅広い生徒に理解してもらえるプログラミング言語であり、優れたファイル入出力機能を備えている。しかし、最近では他言語で非常に強力なコンパイラおよび統合環境がフリーで提供され広まっている。今後、フリーのCOBOLコンパイラや統合環境が登場することを切に願っている。

スクールCOBOL2002の便利な機能

はじめに

今回は、COBOL教育にも役立つCOBOL2002の便利な機能を三つご紹介します。

①COPY文

COPY文は、複数のプログラムに共通のソースを取り込むときに使用します。

同じデータ定義や文をそれぞれのプログラムに記述しなくても、COPYファイルとして定義しておき、それをCOPY文で取り込めばよいわけです。このようにすると、コーディングの量が少なくなるだけでなく、誤りも少なくなります。また、変更が生じたときCOPYファイルを修正するだけで全てのプログラムに反映されるという効果もあり、開発効率がアップします。業務プログラムの開発では、頻繁に使われる機能です。先生方がCOBOL教育をする際、実習に十分な時間がとれないということを何度か耳にしたことがありますが、COPY文を利用すると教育の時間を削減することも可能です。つまり、ファイルのデータ定義だけは先生方がCOPYファイルとして作成しておき、それを生徒に配布してCOPY文で取り込んでもらうというようなことができるのです。生徒はアルゴリズムに専念することができるわけですね。

②入出力ファイルの割り当て

学校の教育において、ファイルは必ず使用すると思います。基本的なことですが、ファイルの割り当てについて触れておきたいと思います。

③プログラムテンプレート

COBOL2002では、一定の書き方をあらかじめ入力してあるプログラムのひな型をプログラムテンプレートとして提供しています。また、COBOL2002が標準提供しているプログラムテンプレートとは別に、プログラマ自身でテンプレートを作成して登録することもできます。よく使うパターンをテンプレートとして登録しておき、これらのテンプレートを有効活用することにより、プログラムの作成を効率よく行うことができます。

1. COPY文

COPY文の使い方

まず、COPY文の使い方について説明します。簡単な機能なので、次の例をごらんになるとすぐに理解できると思います。

＜説明＞

①プログラム1とプログラム2で同じファイル（FILE1）を使用します。

②FILE1のレコードの定義をCOPYファイル（FILE1-REC.cbl）として作成します。

③プログラム1（PROG1.cbl）とプログラム2（PROG2.cbl）は、COPY文を使用してレコードの定義（FILE1-REC.cbl）を取り込みます。

COPYファイル：FILE1-REC.cbl

```
01 FILE1-REC.
   05 支店名          PIC  X（20）.
   05 商品コード      PIC  X（12）.
   05 単価            PIC  9（6）.
   05 売上            PIC  9（10）.
```

プログラム1：PROG1.cbl

```
IDENTIFICATION DIVISION.
PROGRAM-ID.      PROG1.
ENVIRONMENT      DIVISION.
INPUT-OUTPUT     SECTION.
FILE-CONTROL.
     SELECT   FILE1     ASSIGN   TO   SYS010.
     SELECT   PRINT1    ASSIGN   TO   'PRINTER'.
DATA             DIVISION.
FILE             SECTION.
FD   FILE1.
COPY FILE1-REC.   ←―― COPYファイルFILE1-REC.cblの取り込み
FD   PRINT1.
01   PRINT1-REC   PIC  X（120）.
                :
PROCEDURE        DIVISION.
                :
```

プログラム2：PROG2.cbl

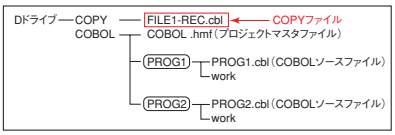
```
IDENTIFICATION DIVISION.
PROGRAM-ID.      PROG2.
ENVIRONMENT      DIVISION.
INPUT-OUTPUT     SECTION.
FILE-CONTROL.
     SELECT   FILE1     ASSIGN   TO   SYS010.
     SELECT   PRINT2    ASSIGN   TO   SYS011.
DATA             DIVISION.
FILE             SECTION.
FD   FILE1.
COPY FILE1-REC.   ←―― COPYファイルFILE1-REC.cblの取り込み
FD   PRINT2.
01   PRINT2-REC.
  05   日付.
    10   年   PIC  X（2）.
    10   月   PIC  X（2）.
    10   日   PIC  X（2）.
          :
PROCEDURE        DIVISION.
          :
```

開発マネージャでのCOPYファイルの操作

COPYファイルを取り込んでコンパイルする方法を説明します。

COPYファイルが次のフォルダにあるものとします。

・Dドライブの「COPY」というフォルダにCOPYファイルを格納し、「COBOL」というフォルダにプロジェクトを作成するものとします。



COPYファイルを取り込んでコンパイルするには、コンパイル時の環境変数「CBLLIB」でCOPYファイルの格納場所を指定します。COPYファイルを取り込むのに必要な作業はこれだけです。

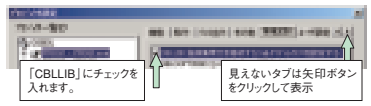
それでは、環境変数の指定方法を次に示します。

①作業中のプロジェクトを「PROG1」にします。

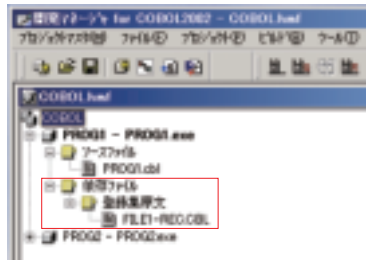
（「PROG1」を選択し、右クリックして「作業中プロジェクトの設定」を選択すると、「PROG1」が太字になり、作業中のプロジェクトとして認識されます。）

開発マネージャのメニューから「プロジェクト」-「プロジェクトの設定」を選択します。

②プロジェクトの設定画面が表示されます。
▶ボタンをクリックして「環境変数」タブを表示し、「CBLLIB」をチェックします。



③環境変数の設定画面が表示されるので、参照ボタンをクリックして「COPY」フォルダを指定し、「OK」ボタンをクリックします。
④「プロジェクトの設定」画面に戻ったら「OK」ボタンをクリックします。開発マネージャの画面に戻ります。
⑤「ビルド」（コンパイル）します。ビルドすると開発マネージャが依存関係を形成します。
「依存ファイル」の左側の「+」ボタンをクリックすると依存関係が形成されたことがわかります。



2. ファイルの割り当て

3種類の割り当て方

スクールCOBOL2002のファイルを割り当て方には、次の三つの方法があります。
①定数指定
SELECT句のファイル名に対して、ASSIGN句で物理ファイル名を直接指定する方法です。文字定数と同様に引用符で囲んで指定します。
②環境変数指定
SELECT句のファイル名に対して、ASSIGN句で「SYS001」というように外部装置名を指定しておき、実行する際に外部装置名に対応する環境変数を使って物理ファイル名を割り当てます。
③データ名指定
COBOLのデータ部で定義したデータ名を指定します。データ名に物理ファイル名を転記してOPEN文を実行します。

割り当て方の詳細

まずファイルの割り当てに必要な基本的な事項について説明し、三つの割り当て方法を説明します。
＜基本的な事項＞
・物理ファイル名は、絶対パスで指定します。
ただし、開発マネージャで作成したプロジェクトのWORKフォルダの下にファイルがあるときは、パス名を省略しファイル名を指定するだけで割り当てることができます。
学校の教育では、パス名を省略した定数指定で割り当てることをお勧めします。
・入力、更新、拡張処理をするファイルは、実行する前にあらかじめファイルを用意しておく必要があります。ファイルが存在しなければオープンエラーになります。しかし、出力ファイルの場合は、ファイルが存在しなくても自動的に生成されます。

＜割り当て方法の詳細＞
①定数指定
次の形式で指定します。
SELECT ファイル名 ASSIGN TO '物理ファイル名'
物理ファイル名の指定例を次に示します。
[例]
'D:¥TEMP¥SAM.dat'
'SAM.dat'　・・・パス名が省略されているので、プロジェクトのWORKフォルダが仮定されます。
②環境変数指定
次の形式で指定します。
SELECT ファイル名 ASSIGN TO 外部装置名
外部装置名は、データ部で定義されているデータ項目の名称と同じであってはなりません。
外部装置名に対応する環境変数の名称は、外部装置名の先頭に「CBL_」を付けたものです。
環境変数の例を次に示します。
[例]
SELECT句：SELECT FILE-01 ASSIGN TO SYS001
環境変数：CBL_SYS001
環境変数に物理ファイル名を割り当てるときは、開発マネージャの実行支援を使用します。
手順1：開発マネージャのメニューから「ビルド」-「実行支援」の順に選択します。すると、実行支援の画面（実行支援 for COBOL2002）が開かれます。
手順2：「ユーザ設定」タブをクリックします。変数名に「CBL_SYS001」を指定し、参照ボタンで物理ファイルを選択するか、「値」欄に絶対パス名を直接入力します。「登録」ボタンをクリックします。プロジェクトのWORKフォルダの下にファイルがあるとき、あるいはプロジェクトのWORKフォルダの下にファイルを作成するときは、「値」欄にパス名を省略できます。
手順3：「ファイル」メニューから「上書き保存」を実行し、実行支援を閉じます。
③データ名指定
次の形式で指定します。
SELECTファイル名ASSIGN TOデータ名
データ名に物理ファイル名を転記してから、OPEN文を実行します。プロジェクトのWORKフォルダの下にファイルがあるとき、あるいはプロジェクトのWORKフォルダの下にファイルを作成するときは、パス名を省略できます。

その他の留意点

ファイルの割り当てに関するその他の留意点を次に示します。
①索引ファイルの割り当て
索引ファイルとその他のファイルでは、物理ファイル名

の指定方法が少し異なります。索引ファイル以外の場合、「拡張子」を含めて指定しますが、索引ファイルは「拡張子」は指定しません。

②索引ファイルの作成

索引ファイルは、OUTPUTモードでオープンしても拡張処理になります。既にあるデータを消去してから作成したい場合は、実行時環境変数CBLISAMDL=YESを指定します。環境変数の設定は、開発マネージャの実行支援を使用して設定（「ファイル」タブの「CBLISAMDL」をチェック）します。

③プリンタファイルの割り当て

プリンタに出力するときは、物理ファイル名として'PRINTER'を指定します。プリンタの各種設定は、実行支援の「設定」-「印刷書式」-「デフォルト」で行います。用紙のサイズや向き、フォント、行間隔、字間値などを設定できます。また、日本語を印刷するときは、印字位置を揃えるために「全角文字と半角文字の間隔を調整する」にチェックを入れます。

3. プログラムテンプレート

プログラムテンプレートの種類

まず、スクールCOBOL2002で用意しているプログラムテンプレートの種類と格納場所について説明します。

《 標準提供されているプログラムテンプレート 》

- COBOL2002プログラム用
  （基本形 / 利用者定義関数 / クラス定義 / インタフェース定義）
- COBOL2002入出力機能プログラム用
  （順 / テキスト / 相対 / 索引 / CSV形式ファイル）
- COBOL2002整列機能プログラム用
  （SORT一括型 / SORT手続き型 / MERGE一括型 / MERGE手続き型）
- COBOL2002報告書作成機能プログラム用
  （報告書作成機能）
- COBOL2002画面機能プログラム用
  （WINDOW節画面機能 / SCREEN節画面機能）
- COBOL2002データベースアクセス機能プログラム用
  （SQL / RDBファイル）

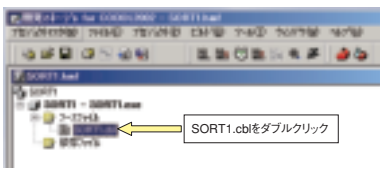
《 プログラムテンプレートの格納場所 》

COBOL2002インストールフォルダの下の¥templateというフォルダ下にあります。

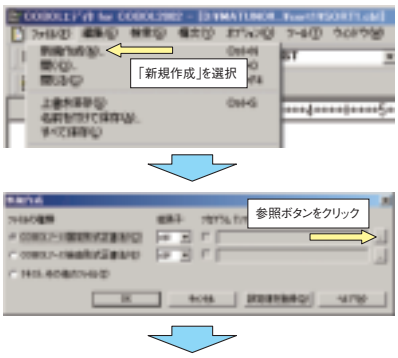
フォルダを意識しなくてもCOBOLエディタのメニューから利用できるようになっています。

プログラムテンプレートを使用してプログラムを作成してみよう

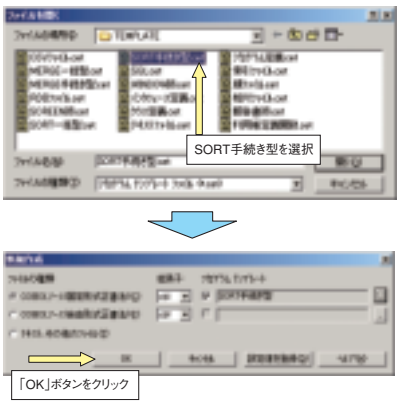
それでは、プログラムテンプレートの利用手順をソートのテンプレートを例に説明します。プロジェクト「SORT1」を作成してあるものとして説明します。



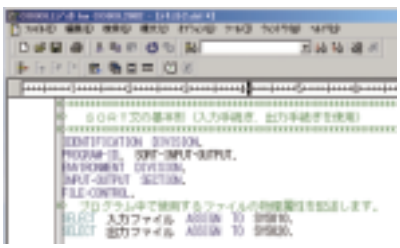
開発マネージャからSORT1.cblをダブルクリックしてCOBOLエディタを起動します。COBOLエディタのメニューバーの「ファイル」メニューから「新規作成」を選びます。すると、新規作成のダイアログが表示されます。



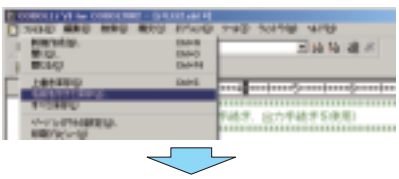
「SORT手続き型」のテンプレートを選択し、「開く」ボタンをクリックします。



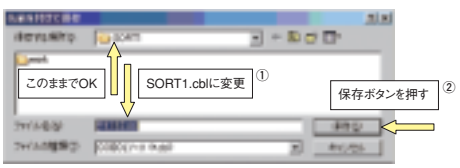
プログラムテンプレートが開かれます。



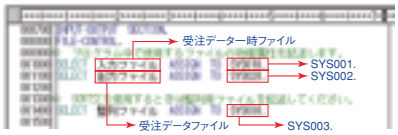
ここで一旦名前をつけて保存します。



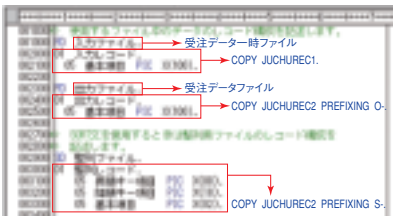
保存する場所はプロジェクトを作成したフォルダが表示されるので、ファイル名を「SORT1.cbl」に変更して保存します。



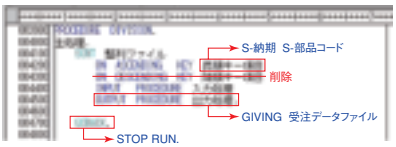
保存ができたらテンプレートを元にプログラムを修正して完成させます。
以下は、プログラムの修正の例です。



レコード定義をCOPY文で取り込むように訂正します。



SORT文や入力手続き/出力手続きなどを訂正して完成させます。



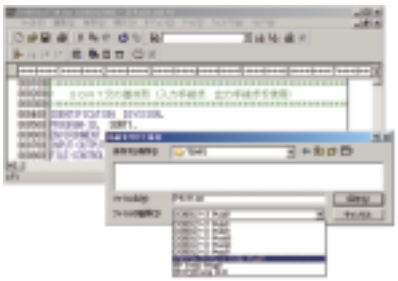
プログラムテンプレートの登録

プログラムテンプレートは、ユーザ自信で登録することもできます。

登録の方法は簡単です。登録したいプログラムをエディタで開きます。

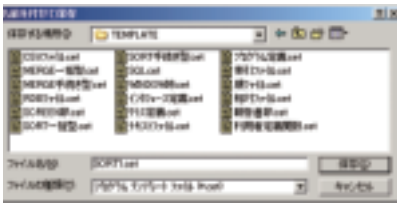
次に「ファイル」メニューから「名前を付けて保存」を選択します。

ファイル名を指定する画面が表示されます。「ファイルの種類」として「プログラムテンプレート（*.cet）」を指定します。



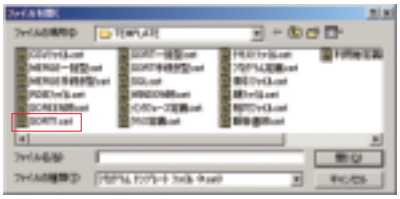
ファイル名を指定し、COBOL2002のインストールフォルダの「TEMPLATE」というフォルダの下に、拡張子「.cet」を指定して保存します（任意のフォルダに保存することもできます）。

※：COBOL2002のインストールフォルダのデフォルトは、次のとおりです。
C:¥Program Files¥HITACHI¥Cobol2002



これでテンプレートとして登録できました。

プログラムの新規作成でプログラムテンプレートを開いてみると登録されていることがわかります。



編集後記

最近、プログラミング言語「COBOL」に追い風が吹いてきています。
(1)「COBOL授業」がわずかですが復活しています。
今年4月から、新たにCOBOL授業を開始する先生から「COBOL授業」を復活させた理由を伺うことができました。
①アルゴリズム学習はCOBOL言語が最適である。
②COBOL言語に関する教材が多くあり授業がしやすい。
③全商1級検定にあるファイル処理をCOBOL言語だと教えやすい。
など「COBOL授業」復活の理由をお聞きすることができました。COBOL言語以外で授業されている先生方、一度、「COBOL授業」を検討していただければ幸いです。
(2)雑誌(月刊ComputerWorld2007年3月号)に「2007年、COBOLは死せず。」が掲載されています。まだまだCOBOL言語は重要なプログラミング言語として生き残っています。
また、COBOL言語を広く普及させ、COBOLユーザの利益を守るために設立された非営利団体「COBOLコンソーシアム」のホームページにもCOBOLの情報が掲載されていますので、是非ご一読ください。（http://www.cobol.gr.jp/）
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.jikkyo.co.jp/index.html)

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.jikkyo.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)5471-2083　http://www.hitachi.co.jp/soft/

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

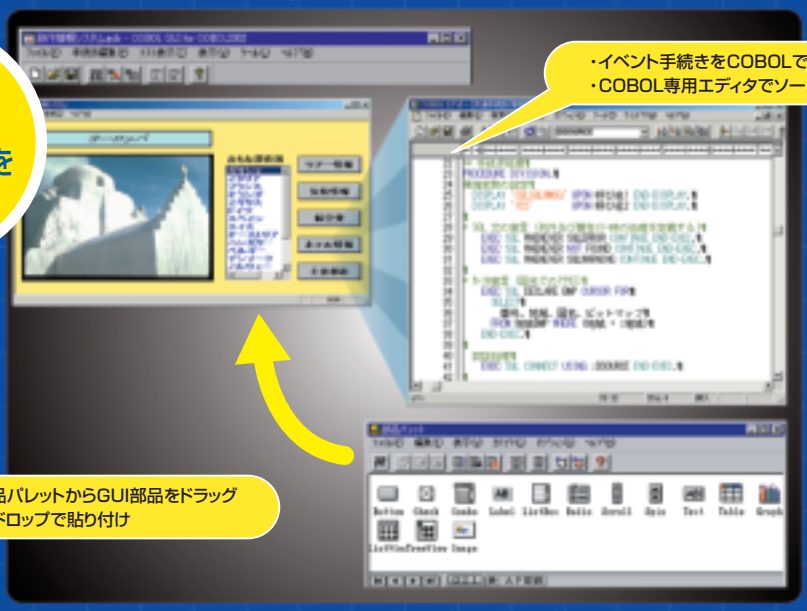
情報処理教育用COBOLコンパイラの決定版

●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



スクールCOBOL2002 GUI構築パック画面

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

PRINTED WITH SOY INK
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。
R100 古紙配合率100%再生紙を使用しています