

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

プログラム言語における 抽象化の歴史

法政大学 情報科学部

コンピュータ科学科 教授 中田育男

COBOL最新企画

Java™とCOBOLの 連携機能ご紹介

ユーザー校事例

高木学園女子高等学校

篠原孝太郎

北海道函館商業高等学校

澤田 耕

企業におけるCOBOL活用

Webシステム構築事例

「スクールCOBOL85」を使いこなそう!

【第3回】COBOLエディタを使いこなそう(3)



All Rights Reserved, Copyright ©2003, Hitachi, Ltd.

Vol. **3**
2002.9

◇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** 改訂版
B5判 272p. 定価 2,000円
2. **システムの開発と運用** 改訂版
B5判 168p. 定価 1,500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

2002年度版 **速攻基本情報技術者 午前**
B5判 216p. 定価 2,200円

合格のための総仕上げ

2002年度版 **速攻基本情報技術者 午後**
B5判 176p. 定価 2,000円

◇初級システムアドミニストレータ試験対策に◇

2002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

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 224p. 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/>

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



法政大学情報科学部 コンピュータ科学科 教授

中田育男 Nakata Ikuo

1960年 東京大学大学院 数物系研究科 修士課程修了
1960年 朝日製作所 中央研究所入社
1977年 東京大学 理学博士
1979年 筑波大学 電子情報工学系 教授
1997年 図書館情報大学 図書館情報学科 教授
1999年 情報処理学会フェロー
2000年 法政大学 情報科学部
コンピュータ科学科 教授
2002年 情報処理学会 名誉会員

主な著書:「基礎FORTRAN」(岩波書店 1991年)「コンパイラ」(オーム社 1995年)「コンパイラの構成と最適化」(朝倉書店 1999年)

プログラム言語における抽象化の歴史

プログラムを書くときに心がけるべき大事なことは、できるだけ一般的な仕様にしておき、再利用ができるようにし、変更に対して強くすること、できるだけ大きなまとまりで理解できるようにすること、などであろう。したがって、プログラムがそのように表現できることを目指してプログラム言語も設計されてきた。より一般的な表現にしたり、より大きなまとまりでものを表現するための手段が抽象化である。抽象化とは、複数の可能性のあるものの、個々の違いを無視して、それらに共通な面を表現することである。プログラム言語の歴史は抽象化の歴史であるともいわれる。コンピュータというものが出現してから現在に至るまでのプログラム言語の歴史を、主として抽象化の観点から考えてみたい。

最初のプログラミングは機械語の命令の列を8進法や10進法の数字を並べた形で書くものであった。これがプログラム言語の第1世代と呼ばれる。次に、数字ではなくアルファベット文字を使って命令語や番地を表現できるようにしたのがアセンブリ言語であり、第2世代言語と呼ばれる。アセンブリ言語でのプログラムでは番地は具体的な数字で表されるのではなく、文字を使ったより

抽象的な表現になる。たとえば、LOOPという名前を付けた番地にジャンプする命令は

JUMP LOOP

のように書くことができ、ジャンプ命令の8進のコードやLOOPという番地が何番地であるかを考えなくてすみ、その番地がずれてもこの命令を書き換えなくてすむようになった。プログラムはサブルーチンというあるまとまった仕事をする単位で作っておき、実行時にそれを適当に組み合わせることも可能になった。

1960年前後に始まった、アセンブリ言語からFortranのような高級言語(機械語のレベルより高級な言語という意味)への進歩は抽象化のステップの中でも最も大きなものであった。普通の数式を書くように式を書けば、それを計算する機械語の命令の列に置き換えられるようになった。このような言語は第3世代言語と呼ばれる。高級言語を使うことによってプログラムは飛躍的に書きやすくなった。最初にFortranを開発した人は、この言語でプログラムを書けばデバッグは必要なくなるはずだと考えたほどであるが、表現が高級になってもそのレベルでのデバッグは常に必要である。

巨大なソフトウェアのデバッグや保守の困難さからくるソフトウェアの危機が認識されて、ソフトウェア工学の必要が叫ばれたのは1968年ころからであるが、ソフトウェア工学における最も重要な考え方は「分割し統治せよ(divide and conquer)」である。大きな複雑なソフトウェアを間違い少なく作っていくためには、それをいくつかの構成要素に分割して、それらを組み合わせて全体を構成するしか良い方法がない。大きなソフトウェアを人間が理解するためには、人間が理解できるような大きさに分割するしかないということである。分割された単位はまとまった意味を持つのがよい。まとまった意味で把握するようにするのが抽象化である。

ソフトウェア工学における最初の重要な提唱は構造化プログラミング(structured programming)であった。その当時はプログラムはまだ手続き主体で考えられており、プログラムのまとまった単位であるモジュールは通常はサブルーチンであり、プログラムはサブルーチンを組み合わせて作られていた。したがって、実行時のプログラムの制御の流れをまず理解することが重要であった。その場合に理解の妨げになるのがgoto文の存在であるということで、goto文を使わずに、より抽象化された制御文(if-then-else文やwhile文など)だけで表現すべきであるとされた。Pascalはそのような機能を備えた言語として使われた。Fortran 66にはそれが不足していたので、Fortran 77が開発された。最近のJava™にはgoto文はない。

分割統治を効果的にするためには、分割されたモジュールの機能が明確であり、各モジュールの独立性が高くなければならない。手続き主体で考えていた当時は、一つのモジュールが一つの関数サブルーチンであるのが良いと考えられた。

もう一つ大事な提唱としては、情報隠蔽(information hiding)があった。これは、手続きを使う

人にはその仕様だけを見せ、実現の情報は使う人から隠せということで、仕様と実現の間に壁をもうけて分割を効果的にするものである。仕様は実現を抽象化したものであり、これは手続き単位の抽象化と言える。これらは1970年ころの話である。

プログラムを理解する、あるいは設計するときは制御の流れだけでなく、データの流を考えることも重要であるが、それまでは、プログラム構造については手続きを中心に考えられ、データは手続きの処理の対象であると考えられていた。しかし、プログラムの理解にはデータの理解が重要であることが認識されるようになってきて、データ中心のプログラム構造が考えられるようになった。データとそのデータに対する基本的な操作をする手続き群とをまとめて考えることで、今までの手続き一つよりもっと大きな単位で抽象化することになる。これは抽象データ型(abstract data type)と呼ばれた。このための機能を備えた言語には1980年前後のClu、Ada、Modula-2、1990年のFortran 90などがある。

抽象データ型ではデータが中心ではあったが、まだ手続きによる操作が主体で考えられており、データはやはり手続きの対象にすぎなかった。その発想を転換して、データが主体であり、手続きはデータ(オブジェクト)が持っているものと考えた言語Smalltalk80が1980年に出現した。このオブジェクト指向言語(object-oriented language)の考え方と、その言語でGUI(Graphical User Interface)がきれいに実現されているのが衝撃的であった。

オブジェクト指向言語では、抽象データ型に相当するものはクラス(class)と呼ばれ、抽象化の単位は抽象データ型よりさらに大きくなって、スーパークラスとサブクラスという関係で結んだ階層構造で一群のクラス(抽象データ型)をまとめて表現することができるようになった。階層構造の上位がより抽象化されたものであり、下位が

それを特殊化したものである。オブジェクト指向言語としてはC++(1990年前後)、インターネットとの関連もあって急速に広まったJava™(1995年から)などがある。

オブジェクト指向言語で抽象化のレベルが上がリ、より大きなかたまりで抽象化可能になったことにより、それを使ってうまい抽象化を行えば、プログラムは分かりやすくなり、その再利用も比較的簡単になる。しかし、うまい抽象化を行うのは必ずしも簡単ではない。プログラムは一般には複数の階層構造を含んだ形で表現されることになるが、それらの間で、機能的にどのような関連を付けたいときにどのような構造にすればよいかは簡単ではなく、試行錯誤を繰り返していい形を見つけることになる。いい形を見つけ、それを出来るだけ一般的な形で表現したものがデザイン・パターンと呼ばれるものである。Java™の基本ライブラリにはデザイン・パターンが多く使われている。それらのデザイン・パターンを理解していればライブラリを理解するのも容易になる。デザインパターン是一群のクラスの階層構造だけでなく、複数の階層構造間の関係にまで抽象化の対象を広げたものと考えることができる。

オブジェクト指向言語をうまく使うことによって、多くのプログラムが以前よりもすっきりと表現できるようになるが、すべてがそうなるわけではない。従来のような手続き指向のプログラムの方が素直な表現になる場合もある。オブジェクト指向言語ではプログラムはクラス単位に作られる。プログラムに新たな機能を追加するのがクラスの追加あるいは少数のクラスの修正ですむのなら、オブジェクト指向言語を使ったことが有効である(そうなるように設計するのがオブジェクト指向言語での設計の仕方である。それが簡単でない場合は、デザイン・パターンなどを参考にする必要がある)。新たな一つの機能の追加のために、多くのクラスにメソッドを追加する必要があるようなプログラムには、オブジェクト指向よりも手続き指向の方が適している。

ある種のデータ構造に対しては、オブジェクト指向の枠組みの中で手続き指向の記述をするためのデザイン・パターンが考えられている。いろいろなクラスのノードからなる木構造のような構造をたどりながらひとまとまりの処理をするプログラムは、各クラスの中でそれをばらばらに表現するよりも、一つの手続きとしてまとめて書いたほうが分かりやすい。すなわち手続き指向の方が適している。Visitorと呼ばれるデザイン・パターンは、このようなプログラムをオブジェクト指向の枠組みの中で表現するためのものである。

プログラムを設計するときにはいろんな側面を考える必要があるから、ある側面だけを考えて抽象化しても良いプログラムになるとはかぎらない。たとえば、基本的な機能面だけを考えて、データの論理的な流れを抽象化してプログラム構造を設計してから、性能を考慮して改良しようとする、もとのプログラムを大きく変更したり、多くのクラスを変更したりする必要があることがある。この場合、性能という側面(アスペクト)は別途記述するというアスペクト指向プログラミング(aspect-oriented programming)も提案されている。コンパイル時に別途記述されたものを取り込むことで、もとのプログラム表現を壊さずにすむ。そのようなことを可能にするためにはその記述法をきちんと決めなければならない。それはまた一つのプログラム言語と考えられる。

このように、複雑なプログラムを理解しやすくするためにさまざまな抽象化が考えられてきたが、歴史的には、より大きなかたまりで抽象化が出来るように言語が開発されてきた。しかし、それですべてが解決されているわけではない。たとえば、オブジェクト指向に対して、デザイン・パターンが考えられたり、アスペクト指向が考えられたりするように、さらに新たな抽象化が考えられている。いずれにしても、大きな複雑なソフトウェアを人間はどう把握すればよいのかという苦闘はいつまでたっても終わりそうにない。

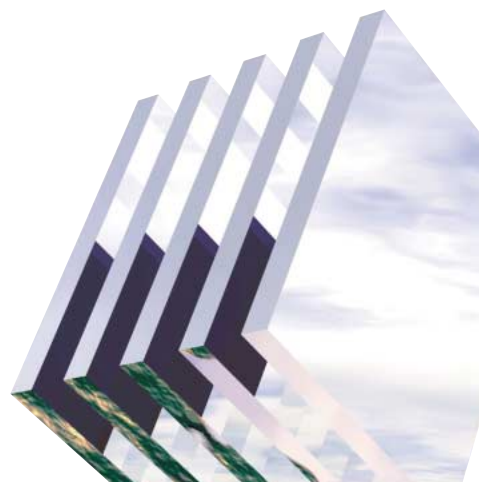
法政大学情報科学部
コンピュータ科学科 教授

中田育男

プログラム言語における抽象化の歴史

Java™とCOBOLの 連携機能ご紹介

最先端のWebシステムでも、
COBOLが活躍しています



はじめに

本格化するe-ビジネスを支えるWebシステムのスピーディな開発のため、Java™をベースにしたWebアプリケーションサーバが普及しています。日立製作所でもCosminexusを提供しています。

Webアプリケーションサーバ環境では、Java™が中心になります。しかし、基幹業務で使用されているCOBOL資産を有効活用することも不可欠です。そのために必要なのがJava™とCOBOLをつなぐ技術です。

本稿では、日立COBOL85ファミリーの「Java™-COBOL連携製品」を例に、この技術について紹介します。

次世代Webシステム構築は Java™とCOBOLの協力で

e-ビジネスを支えるWebシステムは大規模化する一方で、また、e-ビジネスの展開に柔軟かつスピーディに対処する必要もあります。このような状況下で、業界標準のエンタープライズJava™をサポートしたWebアプリケーションサーバが開発基盤として登場しました(図1参照)。Webアプリケーションサーバの環境では、汎用のJava™コンポーネントを活用することによってWebシステムをスピーディに構築することが可能になります。Java™は、ご存じのとおり、「プラットフォームからの独立性」や「インターネットとの親和性」のようなWeb

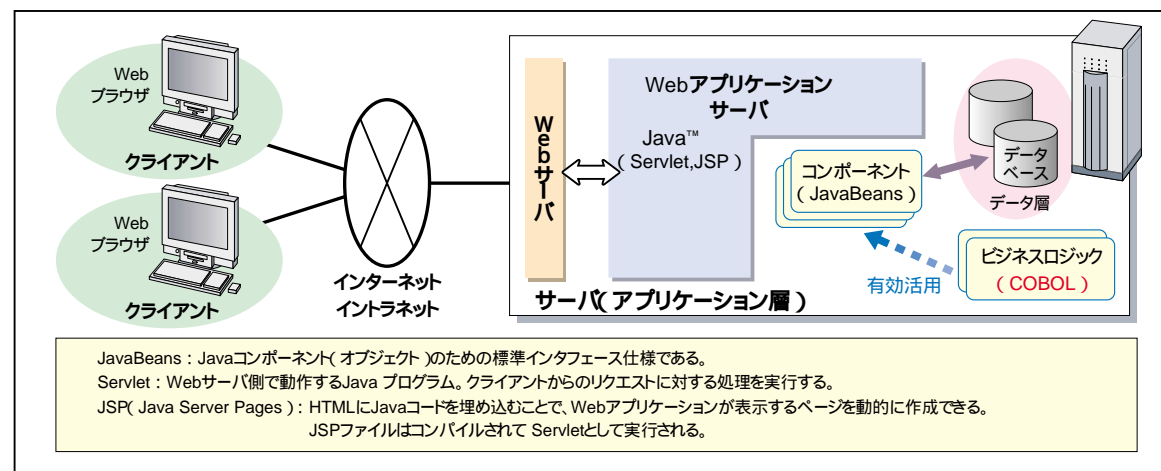


図1 Webシステムの構成

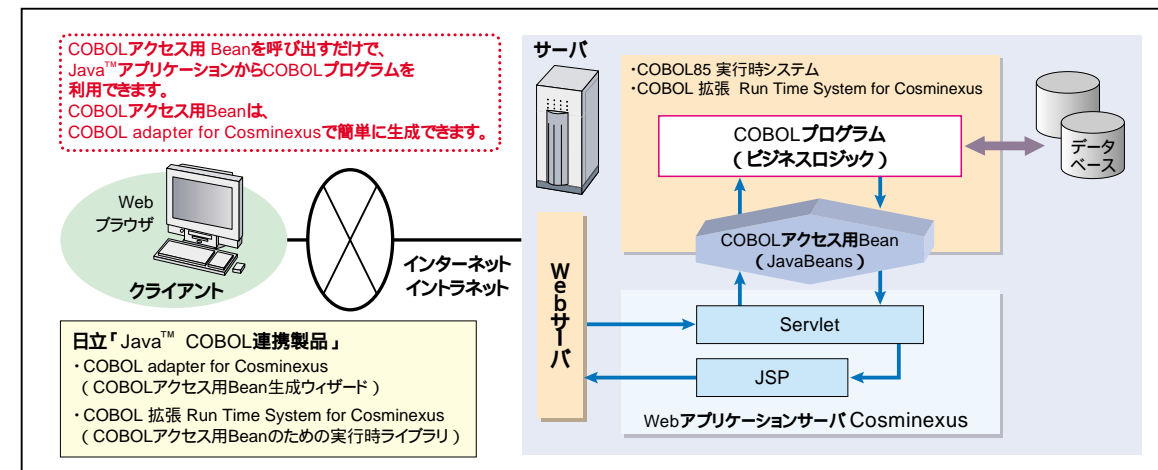


図2 Java™-COBOL連携機能

システム構築に適した特長を備えています。

では、COBOLはどうでしょうか。「インターネットとの親和性」では、Java™に軍配が上がります。しかし、ビジネスロジックの部分では、高精度10進演算や数値文字列操作などのCOBOLの長所が発揮できます。また、使用実績のある既存プログラム資産、開発ノウハウの蓄積や熟練プログラマの数といった観点でも、現状ではCOBOLの方が優位です。多くの企業、団体がかけあえるCOBOL資産およびCOBOLプログラムを考えれば、Java™とCOBOLを適材適所で併用することが現実的であると言えます。特に、高品質なWebシステムを低コストで構築するには、COBOL資産の有効活用が成功の鍵になります。

Java™-COBOL連携機能について

以下、日立COBOL85ファミリーの「Java™-COBOL連携製品」を例にして説明します(図2参照)。

「Java™-COBOL連携製品」の利用によって、Java™とCOBOLの連携が簡単な準備で実現できます。多くのケースで、COBOLプログラムは複雑な会計計算などのビジネスロジックを担うことになります。当然、既に使用実績があり信頼できるCOBOLプログラムがあれば、それらも活用できます。

Java™とCOBOLを仲介するのがCOBOLアクセス用Beanです。COBOLアクセス用Beanを介して、Java™アプリケーションからはCOBOLプログラムがJavaBeansとして見えます。JavaBeansは、Java™アプリケーション(オブジェクト)をコンポーネント化するための標準インタフェースで、基本的にはコンポーネントのデータ(インスタンス変数)の値を入出力するメソッドを規定しています。

COBOLアクセス用Beanは、Java™とCOBOLのデータ型の違いも吸収します。例えば、Java™では文字列型はすべてStringクラスのオブジェクトとなります。Java™からそのまま文字列型の引数が渡されても、COBOLプログラムでは文字列の内容を参照できませんが、必要な変換はCOBOLアクセス用Beanが行います。したがって、Webクライアントからのリクエストによって起動されたServletやJSP(Java Server Pages)などのJava™アプリケーションは、Java™コンポーネントだけでなく、副プログラムイメージで開発したCOBOLプログラムもJavaBeansインタフェースに従って簡単に呼び出すことができます。

「Java™-COBOL連携製品」が提供するウィザードを使えば、COBOLプログラムの連絡節を定義した登録集原文(COPYファイル)からCOBOLアクセス用Beanを簡単な操作で生成できるので、プログラムは使い慣れたCOBOLでのビジネスロジックの開発に専心することができます。

おわりに

COBOLは、基幹業務システム開発で最も活躍しているプログラミング言語であり、常に最新のITに適応してきました。今回ご紹介したJava™-COBOL連携機能は適応の例になります。また、次世代の標準フォーマット定義言語として注目されるXML(Extensible Markup Language)との連携機能も例のひとつです。XML連携機能については先の号(Vol.2 2002.3)で紹介しました。

「多くのプログラマが21世紀にも安心してCOBOLを使い続けられる」を目標にした新国際規格「COBOL 2002」の制定も間近です。今後も、「進化しつづけるCOBOL」にご注目ください。

高木学園女子高等学校

高木学園女子高等学校

篠原孝太郎



本校の概要

本校は、今では考えられないほど女性に対する封建的な重圧の強かった明治41年(1908年)に、「真に自立できる女性の育成」を建学の精神として、女性先覚者・高木 君によって横浜の地に創立された。平成14年9月16日に創立満94年を迎えた歴史と伝統のある女子高校である。卒業生は、平成13年度末で32,500名を数え、産業界はもとより、教育・文化・スポーツ等幅広い分野において、多くの人材を輩出している。

普通科・商業科・情報処理科・家庭科の4学科を設置した、総合女子高校で、今日の日本では数少ない存在となっている。進路状況は、このところ進学志向が強まり65%の生徒が4年制大学・短大へ進学している。

部活動では、ライフル射撃部が創部まもなく3回の全国優勝を果し、高校日本一の実力を有している。ハンドボール部は、県大会でベスト4の経験を持ち、関東大会にも進出。ソフトテニス部は、神奈川県総体3位の経験を持ち、関東大会・全国大会でも活躍している。

実習環境の推移

本校のコンピュータ授業の開始は、昭和58年度より8ビットマシンのNEC8001を25台導入して、商業科の中の「情報処理」科目2単位の実習としてBasic言語から開始した。当初は、何をどこまで教えたらよいのか、とても戸惑いがあったが全商の検定を目標にすることによってその戸惑いもなくなった。

平成4年4月から情報処理科を設け、1年生1クラスを募集し、情報システムの開発に必要な専門的な知識・技術を習得し、将来、システムエンジニア・プログラマなどのスペシャリストとして活躍できる人材を育成することを目標にスタートした。1年次には、科目「COBOL」6単位、「情報概論」4単位を配当し、全商情報処理検定2級レベルの内容を到達目標としている。平成13年度より科目「COBOL」、「を科目「プログラミング」に変更した。この年(平成4年)からCOBOL実習を導入し、DOS版COBOLを使い生徒達に実習をさせ始めた。その後、何年かたちコンピュータの更新の際に、スクールCOBOL85を導入し、Windows®化を果たした。現在、コンピュータ教室は3教室有り、そのうちの1教室50台にスクールCOBOL85をインストールし、実習に使用している。

コンピュータ実習室の概要

本校のコンピュータ実習室は、3ヶ所あり、ラーニングラボラトリー、コンピュータ教室A、コンピュータ教室Bと名づけてある。そのうちラーニングラボラトリーには、平成12年4月に機種を変更し、OSはWindows®98を導入し、クライアント/サーバシステムを構築した。この機会を利用してWindows®版COBOL85コンパイラを使用することにした。

Windows®版COBOL85コンパイラは、GUI環境でのCOBOL用のエディタやデバッガが付属し、COBOLの予約語の色分け機能や自動シーケンス機能(行番号の自動付加)、正書法による領域・境界が補助線で分かりやすく表

示されており、コーディングからスムーズにタイピングができ、デバッガによるデバッグやデータのトレースも容易になり、ネットワーク環境も手伝ってより効率的な実習を行えるようになった。生徒も慣れてくるととても早くコーディングから構文チェック、ビルド、実行とスムーズに実習ができるので好評である。

コンピュータ教室Aは、平成13年4月に文部科学省の半額補助を受け、NECMateNX45J50台とサーバ機としてEXPRESS5800を1台、カラーレーザープリンタ1台を導入した。PCSEMIGを入れ画像転送やリモート操作が可能なシステムを構築してある。

COBOL実習の概要

本校のCOBOL実習は、情報処理科の1年次より始め、全商情報処理検定2級の範囲をマスターして合格を目標としている。従って、4月の1ヶ月間は、教室でアルゴリズムやCOBOLの文法についての講義を行う。その後実習と講義と交互に行い、ページコントロール、グループタイトル、テーブル(配列)、逐次探索(線形探索)等とマスターさせていく。検定が終わった後、検定に出題された流れ図と言語問題を実際にコーディングさせて実習させる。出題された問題が、実際、正しく動くかどうかを試してみようということで、クラスをグループに分け、提出締切日を指定して実習させる。出来るだけ自分たちで考えさせることを主眼に、余り質問も受付けないようにした。これをやらせることによって1年間の総復習となり、しかも大変力がつくことはき面である。このことが更に次のステップにつながり、1級から基本情報処理技術者試験へと挑戦していくことになる。また、実習は、時間がかかるがメリットもたくさん有る。特に、デバッグすることで力がつくし、実行してうまく行けば感動も意欲もわく。

2年次では、全商情報処理1級に挑戦させるため、4月当初より二次元テーブルのイメージトレーニング、特に添え字の学習に力を注ぐ。非逐次探索(二分探索)、順位付け、内部分類、ファイルメンテナンス等の学習へと続く。実習と講義を交互に入れながら、生徒が講義にあきないように実習をしていく。1級には6割の合格者を毎年出す。

COBOL導入について

情報を学習することは、パソコンの操作を覚えることが目標ではなく、パソコンを活用して適切な情報を得ることの学習が目標である。そのひとつとしてアルゴリズムをマスターすることではないか。そのためにソフト活用教育のみならず、言語教育も取り入れて学習して行くことが必須と考える。

たくさんあるコンピュータ言語のうち、COBOLを選んだ理由は、文法が複雑でなく分かり易いこと、高校生でも理解できる英文であること。また、全商情報処理検定にCOBOL言語を採用していることや経済産業省の基本情報処理技術者試験でCOBOLの問題を選択して受験できること。なかでもスクールCOBOL85のコンパイラが導入できたことである。

COBOL実習年間指導計画表

実施時期	主な指導内容	
	1年次	2年次
5月	集計、合計、平均	二次元テーブル
6月	条件判断(IF文) 最大値、最小値	非逐次探索
9月	ページコントロール	順位付け
10月	グループトータル	内部分類
11月	一次元テーブルを 使った集計	〃
12月	線形探索	ファイル処理

情報処理科1年「プログラミング」5単位

情報処理科2年「プログラミング」3単位



北海道函館商業高等学校



北海道函館商業高等学校

澤田 耕

学校の概要

本校は、明治19年に創立され、現存の北海道の公立学校では最古の存在であり、また教育目標である「個人としての資質の育成」「社会人としての資質の育成」「産業人としての資質の育成」を胸に巣立った卒業生はすでに32,000名を超える輝かしい歴史と伝統に支えられた商業高校である。

学校規模は、全校19クラスあり、学科は国際交流能力を高め国際社会に対応できるスペシャリストを育成することを目標とする「国際経済科」、また情報通信ネットワークの活用を図りビジネス活動の専門的な情報を処理・活用できるスペシャリストを育成することを目標とする「情報処理科」、そしてビジネスに対する望ましい心構えや理念を身につけビジネスの諸活動のスペシャリストを育成することを目標とする「流通ビジネス科」の3学科を設置し、より専門性を深め変化する社会に対応できる能力と豊かな人間性を育てることを目指している。

進路においては、就職は地元企業の事務職を中心に男女とも高い就職合格率を上げている。また、進学についても大学・短大をはじめ年々増加傾向にある。

情報処理科の概要

本校の情報処理科は、1年次では「情報処理」「プログラミング」併せて6単位を設定し、言語学習と表計算ソフト（Excel）を中心に指導している。特に言語学習（COBOL）においてはプログラミングの知識や技術を深めさせること

を目標としている。更に3学期からはこれに基本情報技術者試験に対する学習が追加される。1年次の検定・資格目標としては、全商「情報処理検定」2級、全商「コンピュータ利用技術検定」3級を目指す。2年次では「プログラミング」7単位を設定し、言語学習(COBOL)においてはより専門的な学習を行う。また、基本情報技術者試験に対しても多くの時間を配当している。検定については全商「情報処理検定」1級を目標としている。3年次には、1、2年次で学習した言語学習を基礎として、「総合実践」「情報管理」併せて5単位の中でモデル企業を設立し、各生徒が先生の指導のもとにシステム設計を行い、模擬的運用を通して経営活動を主体的に行うための能力と態度を育成することを目標としている。また、基本情報技術者試験の合格を目指す。

実習環境の概要

本校の実習環境については、平成11年の校舎改築に伴い施設設備も大幅に充実し、十分な実習環境が整った。COBOLの実習においても従来のDOS版からGUIを利用可能なWindows®版(スクールCOBOL85)へと変更した。このスクールCOBOL85の導入により、テキストの色分けや構文チェックなどのエディタの使いやすさ、DOS版に比べ操作性に優れているなどの点から実習効率の向上を図ることができた。

今後は、本校のネットワーク環境のなかで、さらに効率よく実習を行なうためには、どのようにすればより開発効率が向上できるかが課題である。

情報処理科実習室の概要

サーバ	富士通 PRIMERGY B225	1台
PC(教師用)	富士通 FMV-6000SL	1台
PC(生徒用)	富士通 FMV-6000CLs	41台
プリンタ(教師用)	カラープリンタ EPSON PM-890C	1台
プリンタ(生徒用)	レーザープリンタ NEC Multi Writer 125	4台
スキャナ(教師用)	EPSON GT-9700F	1台
スキャナ(生徒用)	EPSON GT-8200U	41台
教材提示用モニタ	Mitsubishi RDS151X	21台

本校のプログラミング実習手順と年間計画

段 階	段 階 (実 習 室)	段 階 (実 習 室)
問題の分析	COBOLエディタの起動	COBOL開発マネージャの起動
データシートの作成	実習データの入力	プロジェクトマスタの作成
スペーシングチャートの作成		ソースプログラムの入力
メモリーマップの作成		エラー修正
流れ図の作成		ビルド(コンパイル)
コーディングシートの作成		実 行
		実行結果印字
		コンパイルリスト印刷
		完成・提出・検印

1・2年次実習の年間計画

	1 年 次	2 年 次
4月		内部整列
5月	データの入力と出力	2次元テーブル
6月	四則演算・合計と平均	2次元テーブル・データチェック
7月	見出しの編集・条件の判定	ファイル処理(整列・併合)
8月	最大値と最小値・データの集計	ファイル処理(照合・保守)
9月	複合条件・条件名条件・多方向への分岐	ファイル処理(照合・保守)
10月	ページコントロール・グループトータル	索引編成ファイル・相対編成ファイル
11月	1次元テーブル・逐次表引き	総合実習問題
12月	全商2級検定対策	全商1級検定対策
1月	全商2級検定対策	全商1級検定対策
2月	非逐次表引き	基本情報技術者試験対策
3月	順位づけ	基本情報技術者試験対策

Webシステム構築事例

当該事例では携帯電話用顧客管理システムを紹介する。

本システムでは携帯電話端末への残高補充 / 期限管理を

お客さまセンターのWeb 画面から行える環境を実現し、業務効率化と顧客サービスの向上に役立てている。

システムの中核には日立製作所のWebアプリケーションサーバ「Cosminexus」(コズミネクス)を採用。

基幹システムのCOBOLプログラムと

Java™アプリケーションとの連携を、スピーディかつ低コストに実現している。

気軽に使える携帯電話を提供

今や我々の生活にとって欠かせない存在となった携帯電話。ビジネス、パーソナルの別を問わず、あらゆる場面で広く活用されている。端末の高機能化も一段と進展。Webアクセスやメール機能はもちろんのこと、最近ではCCDカメラや強力なサウンド機能を搭載した製品も登場している。

携帯電話のユーザー層や利用シーンが大幅に拡大したことで、「もっと手軽に使いたい」「一時利用に使える携帯が欲しい」といった新たなニーズも生まれてきた。そこで、使いすぎなどを気にすることなく、安心して携帯電話を活用できる環境を提供することに着目した。

*

お客さまセンター用の 顧客管理システムを Java™アプリケーションで構築

残高や使用期限の確認、補充処理は、携帯電話端末から行うのが一般的なパターンである。しかし時には残高・使用期限に関する問合せやクレームが、直接お客さまセンターに入るケースも考えられる。そこで顧客管理システムの再構築を実施、オペレータが問合せへの対応や補充

処理などをWeb画面から行える環境を実現した。

本システムはJava™によるWebアプリケーションで構築されたが、この理由としては次の要因があった。

お客さまセンターには既にWebベースのシステム環境があったので、これとシームレスに統合したいと考えた。クライアント・サーバー型のシステムと異なり、Webアプリケーションならクライアントにプログラムをインストールする必要がないというメリットもある。

ただし、ここでは解決すべき課題もあった。既存システムはCOBOLで構築されていたため、Java™-COBOL連携が必須となったのである。この問題を解決すべく、選ばれたのが日立のWebアプリケーション「Cosminexus」である。長年にわたって蓄積された基幹システムにおける高度なCOBOL開発技術と業務ノウハウを活用し、かつ短期でWebアプリケーションを構築するにはCosminexusが最適だと考えた。

*

Java™-COBOL連携を スピーディに実現

今回のシステムでは、2つの分野でJava™-COBOL連携が行われている。一つは「交換機業務との連携」、もう一つは「共通サブルーチンとの連携」である。

残高補充処理は携帯電話端末から行えるため、通常は電話交換機とCOBOLで構築された業務アプリケーションとの間だけで処理が行える。

しかし先にも述べたように、イレギュラー処理やクレーム処理などの際にはお客さまセンターのWeb画面から補充処理作業を行うため、Java™で構築されたWebアプリケーションとCOBOLの業務アプリケーションを連携する必要がある。

Java™とCOBOLの連携部分を一から開発するとなると、それなりの工数を覚悟しなければならない。しかし、日立製作所が提供しているCOBOL adapterを使えばJava™-COBOL連携モジュールなどを自動生成することが可能。これにより開発スピードとシステム品質の両方を高めることができる。

また後者の共通サブルーチン処理との連携については、残高の変更履歴管理や期限を過ぎたカードの残高没収処理、経理データ作成処理など、Java™側とCOBOL側で重複する処理が発生する。同じ処理を別々に行うのは無駄なので、こうしたものについてはモジュールの共通化を図った。ここでも実際の処理を行うサブルーチンは既存の開発資産が活かせるCOBOLで統一。COBOL adapter

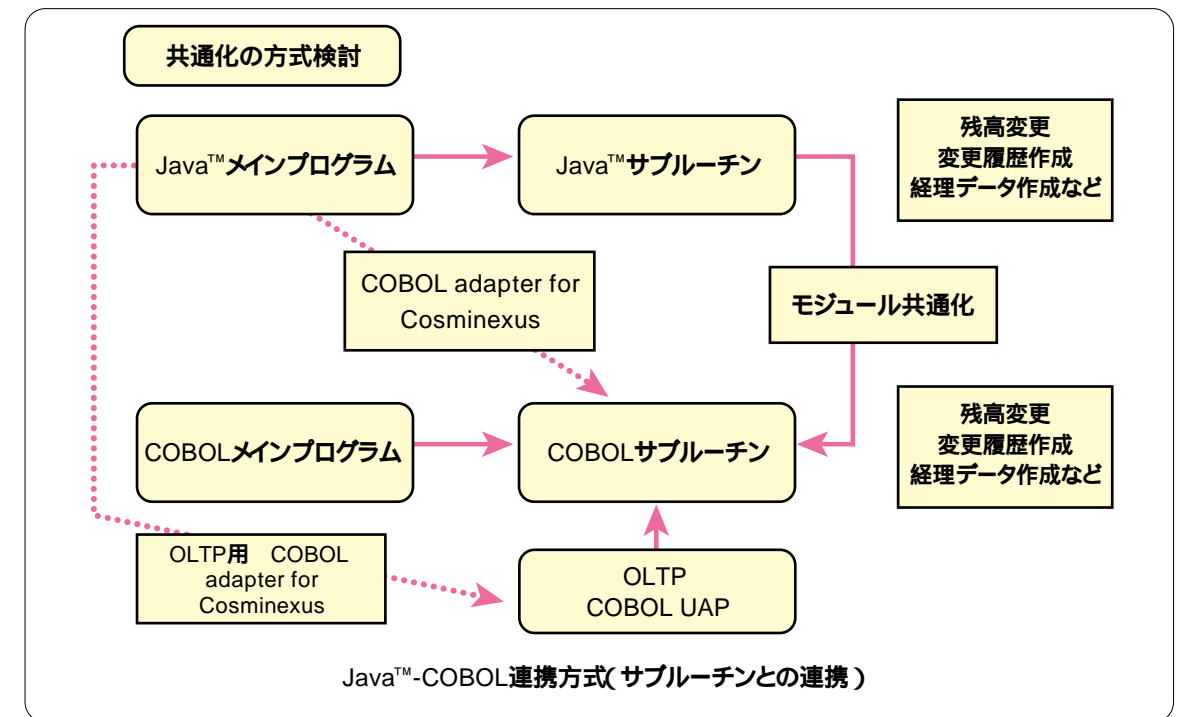
を利用し、サービス提供を行うCOBOLプログラムを経由して必要なサブルーチンを呼び出す方式を採用した。

*

既存のシステム資産を 新たなサービスにも活かす

携帯電話という重要な社会インフラを担うシステムだけに、高度な性能・信頼性も要求される。しかし、サービス開始以来、システムはトラブルなしで稼働。本システムの構築期間はわずか4ヶ月。Cosminexusでなければ、これほどの短期構築は実現できなかった。

これまで蓄積してきたCOBOLプログラム資産を活かしつつ、Java™ベースでの新たなサービスを実現することができた。この取り組みは、レガシーシステムも含めたシステム統合を検討している企業にとっても、大いに参考になることであろう。



第3回

COBOLエディタを使いこなそう(3)

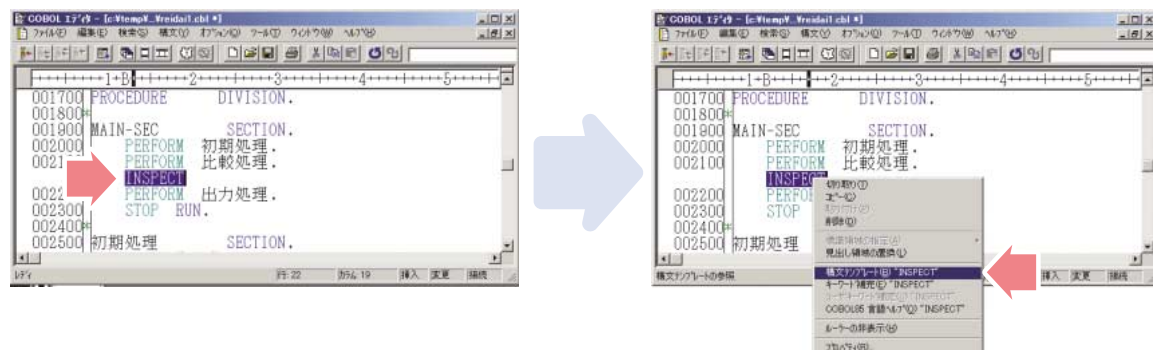
今回は、COBOLエディタの構文テンプレートとオンラインマニュアルについてご紹介します。

1 命令文の構文が不明確なとき、すぐに構文を参照することができます。

・INSPECT命令の構文が不明確な場合

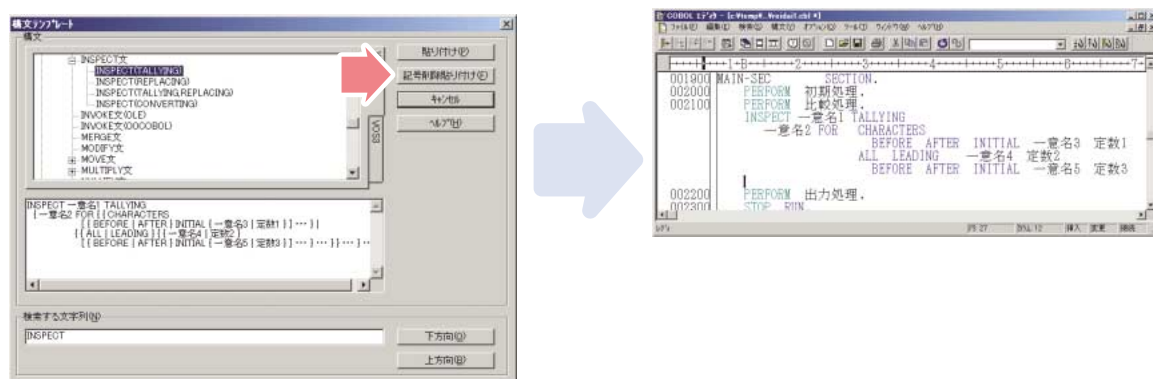
方法

エディタ画面の記述したい箇所にキーワード「INSPECT」を入力します。
入力した「INSPECT」を右クリックし「構文テンプレートの「INSPECT」」を選択します。



方法

以下のような構文テンプレート画面が表示されます。
例えば「INSPECT(TALLYING)」をクリックすると、プレビュー画面に構文が表示されます。
「記号削除貼り付け」をクリックすると記号が削除されて該当する箇所に貼り付けられます。

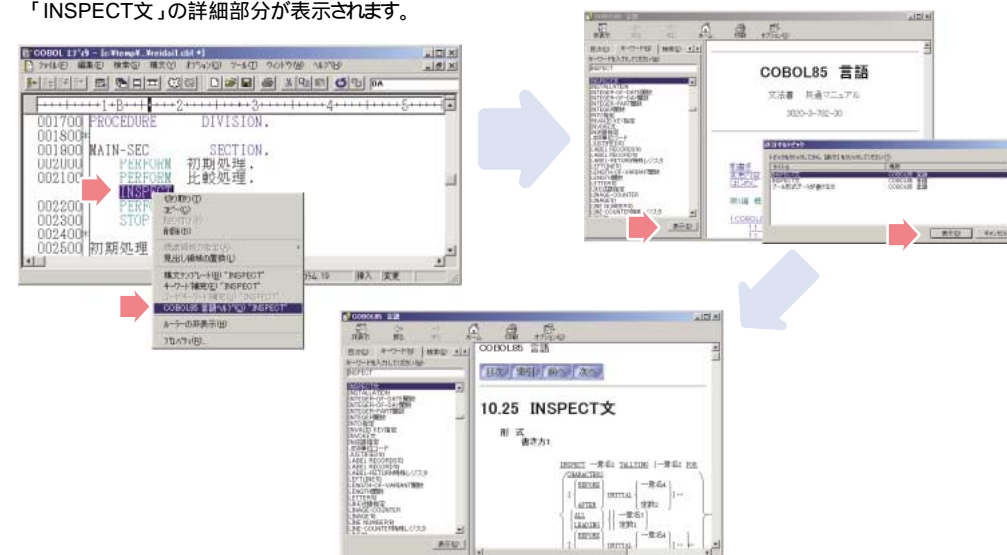


2 エディタ画面よりオンラインマニュアルが参照できます。

・INSPECT命令のオンラインマニュアルを参照する場合

方法

入力した「INSPECT」を右クリックし「COBOL85言語ヘルプの「INSPECT」」を選択します。
「COBOL85 言語」オンラインマニュアルの画面が表示され、
「表示」をクリックすると「該当するトピックス」画面が表示されます。更に「該当するトピックス」画面の「表示」をクリックします。
「INSPECT文」の詳細部分が表示されます。



(注)本紹介内容は、最新スクールCOBOL85のCOBOLエディタを使用しています。

編集後記

本レポート発行も3回目となり、早や1年になりました。皆様のお役に立っているか心配しながら、今回も無事発行することができました。

さて、今回までのCOBOL教育実践例の実習内容を再読してみると、しっかりと目標を定め、工夫された学習内容がよくわかります。「全商情報処理検定2級」から「1級」更に「基本情報処理技術者試験」へとステップアップし、「初級システムアドミニストレータ」にも合格者を出されている

学校もある。(スクールCOBOL85も微力ながら貢献している、と自己満足している次第です。)

一時、COBOL離れが言われておりましたが、最近では「アルゴリズムの基礎」を身に付けるために新しくCOBOL実習を開始される学校が増えてきております。新しくCOBOL実習を始められる先生方に安心していただくためにも、今後もより良いレポート発行を目指し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.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B館
Tel.(03)5471-2592 <http://www.hitachi.co.jp/soft/>

掲載されている会社名、製品名は、それぞれ各社の商標または登録商標です。

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」をご導入されている場合、
最新の「日立スクールCOBOL85」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COBOL85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COBOL85 GUI構築バック画面

情報処理教育に最適。

多くの企業で使用されている実績を誇るCOBOL85コンパイラをベースに開発されたのが、
スクールCOBOL85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COBOL85

最新のCOBOL基準に準拠しています。
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COBOL85 GUI構築パック

従来のCOBOL言語仕様に若干の言語仕様を加えるだけで、
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COBOL85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(税別)

スクールCOBOL85

1セット	58,000円
21セット	940,000円
41セット	1,725,000円

スクールCOBOL85 GUI構築パック

1セット	88,000円
21セット	1,408,000円
41セット	2,584,000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