

# Java プログラミング

ポリテクセンター関西 秋田 正秀  
(関西職業能力開発促進センター)

## 1. はじめに

関西職業能力開発促進センターでは3年ほど前からJavaプログラミングに関する能力開発セミナーを行っています。Java言語は当初、アプレットの作成で一躍脚光を浴びました。それから時が経つにつれ普及が進み、現在では多くの方に使われているプログラミング言語の1つになっています。Java言語で開発できる内容はアプレットはもちろんのことアプリケーションやBeans, Servletなど多岐にわたります。ちなみに、Java言語はSun Microsystems社が開発したオブジェクト指向言語です。

本内容では関西職業能力開発促進センターで行っているJavaプログラミングに関する能力開発セミナーの実践報告と、合わせて簡単なJava言語の紹介を行います。関西職業能力開発促進センターで現在実施しているJavaプログラミングに関する能力開発セミナーはJavaプログラミング導入技法, Javaプログラミング実践技法, Javaプログラミング応用 (JavaBeans編), Javaプログラミング応用 (Swing+JDBC編), Java Servletの5種類です。

## 2. Java 言語

ここではJava言語の概要について紹介します。

### 2.1 Java 言語の起源

Java言語はSun Microsystems社によって開発されました。1991年にSun Microsystems社が行ったテレビ、ビデオデッキ、トースター等の家電製品用のソフトウェア開発プロジェクト(グリーンプロジェクト)の中からJava言語は生まれました。Java言語を開発した中心人物はJames Goslingという方です。この方はこのプロジェクトでC++を用いて低コストでハードウェアに依存しないソフトウェアプラットフォームの開発を行っていました。その後、さまざまな理由から使われていたC++が外されて代わりにOakという言語が開発され、それがこのプロジェクトに使用されることになりました。Oak言語はC++をベースにしながらC++の多重継承や自動的な型変換、ポインタを排除したものです。しかし、この家電製品用ソフトウェア開発プロジェクトは発展せず、副産物としてこのOak言語が残りました。Webが出現した1994年にOakはJavaと名前を変更されました。その後、JavaはWebRunnerと呼ばれるWebブラウザのためのベースとして使われました。このWebRunnerの良さが実証されてJava/HotJava計画につながりました。そして、1995年春のSunWorld EXPOでJavaはブラウザであるHotJavaとともに発表され、爆発的な広がりを見せました<sup>1)2)3)</sup>。ちなみに、HotJavaはSun Microsystems社がJava言語を使って開発したブラウザです。画期的だったのはブラウザの中でJavaプログラムが動作することに加えて、Javaプログラムがダウンロードできることでした。もともとブ

ブラウザではHTML言語で記述されたファイルがロードされてブラウザで実行され静的な内容を表現していました。一方、HotJavaではその中でJavaプログラムを動作させることができました。これはブラウザの中で動的な表現ができることにつながり、さまざまな使い方が考えられます。また、Javaプログラムがダウンロードされるということは、サーバにJavaプログラムをおいておけばアクセスしに来たクライアントがダウンロードすることになり、クライアントはあらかじめJavaプログラムを持っていなくてよいこととなります。要するにクライアント側の負担が軽くなることを意味します。このことはThinクライアントと呼ばれています。結局はJavaプログラムが動作するブラウザがあれば、どのコンピュータでも動的な表現ができることとThinクライアントを実現できるということになります。

## 2.2 Java言語の特徴

Java言語にはいくつかの特徴があります。ここでは主な4つの特徴を紹介します。

1つ目の特徴としてプラットフォーム独立があります。図1に実行コード生成過程を示します。一般にC言語やC++言語などでは各コンピュータにコンパイラがあって、ソースコードをコンパイルしてコンピュータ用のバイナリコードを生成します。もっと言えば、そのコンピュータに入っているCPUが解釈できるコードが生成されます。例えば、身近なコンピュータではWindowsが動作するDOS/V機があります。DOS/V機にはCPUにPentiumが使われています。Windows上でC言語のソースコードをコンパイルするとCPUのPentiumが解釈できるバイナリコードが生成されます。このバイナリコードはPentiumが使われているDOS/V機では動作しますが、UNIX機やMacintoshで動作させようとしてもCPUが違うので動作しません。一方、Javaの場合はというとJava仮想マシンがあればどのコンピュータでも動作します。Javaの紹介でよく使われていた言葉に「Write Once, Run Anywhere」があります。これはJavaプログラムを一度作成すればどのコンピュータでも動作しますよということ

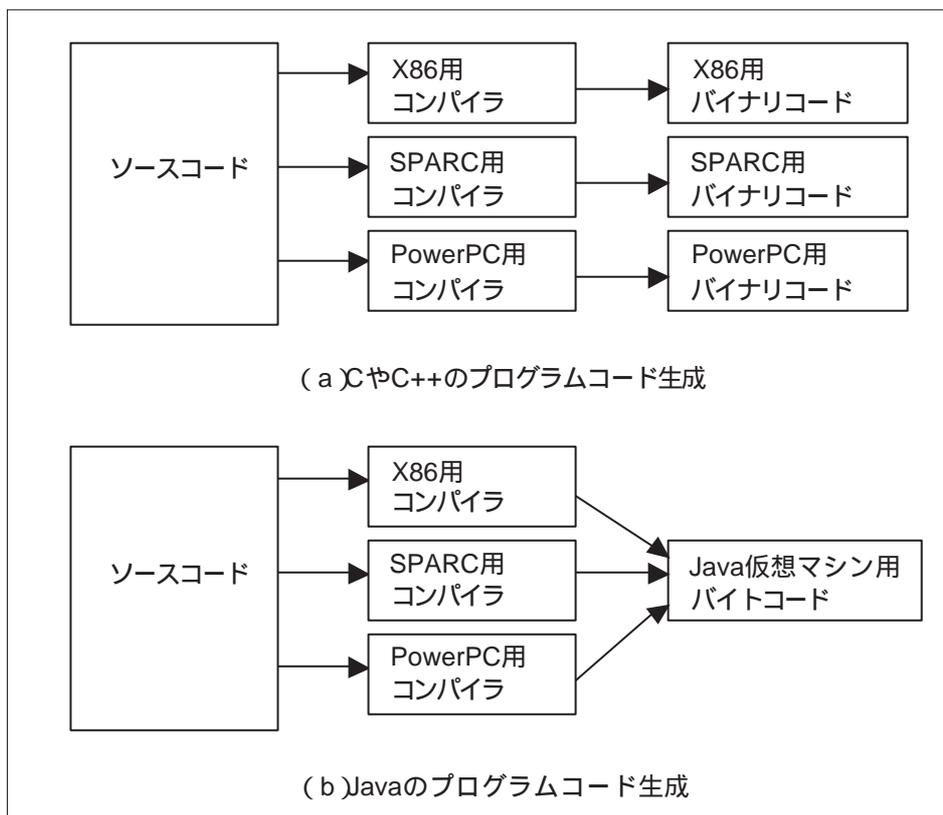


図1 実行コードの生成

ず。Javaではソースコードを作ってコンパイルを  
するところまではC言語やC++言語などと同じで  
すが、コンパイルして生成されたコードが違います。  
生成されるコードはCPUにネイティブなコードで  
はなく中間コードです。一般的にはバイトコードと  
言われています。それで、生成されたバイトコード  
はJava仮想マシン上で実行されます。Java仮想マ  
シンが各コンピュータに実装されていれば生成され  
たバイトコードはどのコンピュータでも動作するこ  
とになります。ある意味、機種差分をJava仮想マ  
シンが吸収していることになります。

2つ目の特徴としてオブジェクト指向がありま  
す。はじめの方でも紹介しましたが、Java言語は  
C++言語をベースに開発されています。C++言語  
はオブジェクト指向言語でオブジェクト指向の要素  
を持っています。例えば、カプセル化や多態性、継  
承などです。これらの特徴はJava言語でも取り入  
れられています。カプセル化は修飾子を使って可視  
の範囲を決めることができ、結果としてインスタ  
ンスをブラックボックス化することができます。多態  
性についてはオーバーロードまたはオーバーライド  
機能で同じシグニチャで内容を切り替えることが  
できます。継承については存在するクラスをベースに  
クラスを拡張する内容です。C++では多重継承で  
きるのに対しJavaでは単一継承となっています。  
Javaでは多重継承の代わりにインターフェイスを  
使って補っています。

3つ目の特徴はポインタが無いということです。  
C言語やC++言語ではポインタやポインタ演算機  
能があります。ポインタは理解するのが難しくこれ  
ら言語習得の難関になっています。実際にポインタ  
を扱ったプログラムを記述するとコンパイルはうまく  
いくが動作が保証されていないということなどよく  
あります。Javaにはポインタがありません。で  
すから、プログラムもシンプルで作りやすく、加え  
てメモリリークを防ぐ機構なども用意されていま  
す。

4つ目の特徴はインタープリタ言語であるとい  
うことです。C++言語はコンパイラ言語ですが、  
Java言語はインタープリタ言語です。コンパイラ

言語はCPUにネイティブなコードを生成し直接実  
行することができるので実行速度が速くなります。  
一方、インタープリタ言語は1行ずつコードを読み  
込んで解釈し実行していくので、実行する速度がコ  
ンパイラ言語に比べて遅くなります。特に注文をつ  
けられているのがSwingです。Swingを使ったプロ  
グラムは特に時間がかかります。対策としてはこれ  
までは実行速度を上げるためにJIT (Just-In-Time)  
コンパイラが使われたり、最近ではJavaHotSpotと  
いう技術が導入され実行速度を上げています。

その他、安全性やマルチスレッド対応など豊富な  
特徴をJavaは持っています。

### 2.3 開発キット

Javaにはプログラム開発用のキットがあります。  
図2にJava 2 SDK1.3をWindowsにインストール  
したときの状況を示します。Javaプログラムを作  
成するにはベースとしてJDK (Java Development  
Kit) が必要です。各OSに対応したJDKがありま  
す。JDKは頻繁に更新されており使っているとす  
ぐに新しいバージョンが出てきます。その意味から  
考えると安定しているとは言えません。バージョン  
はJDK1.0.2, JDK1.1.X, そしてJDK1.2.Xと頻繁に  
更新され、今ではJDK1.3になっています。ちなみ  
にXは番号でJDK1.1のときは1から8まで、  
JDK1.2では1から2までありました。細かいところ  
ではJDK1.2.2\_001などというバージョンもあり  
ました。現在ホームページではJava 2 SDK1.3とい  
う名前になっています。JDKにはJavaコード開発  
のためにいくつかのツールが入っています。インス  
トールされたフォルダの中でJavaコード開発のた  
めのプログラムはbinフォルダに入っています。bin  
フォルダにはソースコードからバイトコードを生成  
するコンパイラのJavac.EXEや、Javaのバイトコ  
ードをOSで実行するためのJava.EXE、アプレット  
のテストコンテナとして使われるAppletviewer.EXE  
などがあります。Java.EXEはOSでJavaのバイト  
コードを実行するプログラムなので、このプログラ  
ム自体がJava仮想マシンということになります。  
また、アプレットを動作させる場合はJavaのバ



図2 Java2 SDK インストール

イトコードと合わせてHTML言語で記述されたファイルが必要になります。Javaに対応したブラウザやAppletviewer.EXEはHTML言語で記述されたファイルを読み込み、その中に「<APPLET>」のタグがあったらブラウザに内蔵されているJava仮想マシン上でアプレットを実行させます。

その他のツールとしてデバッガのjdb.EXEやソースコードからドキュメントを作成するためのjavadoc.EXE、アーカイブファイルの作成、解凍を行うjar.EXEなどがあります。

## 2.4 Javaプログラムの開発

Javaプログラムには先ほどアプリケーションとアプレットの2種類があることを紹介しましたが、

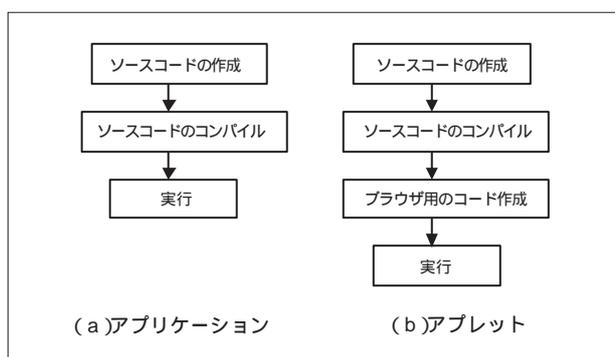


図3 バイトコードの実行

ともにバイトコードを生成するところまでは同じ手順を踏みます。図3にバイトコードを実行するまでの過程を示します。

実行するときにはアプリケーションの場合はJava.EXEを使って実行しますが、アプレットはHTML言語でブラウザ用のファイルを作成してからAppletviewer.EXEまたはブラウザから実行します。HTML言語で作成するファイルはホームページを作成するために使われており、周知のとおりです。ブラウザやAppletviewerでアプレットを実行させるためには「<APPLET>」というタグを記述します。ブラウザやAppletviewerはこのタグを見つけると内蔵されているJava仮想マシン上でこのタグで指定されたクラスファイルをロードし実行しようとしています。ブラウザがJavaに対応していない場合はこのタグは無視されます。

## 3. セミナーについて

ここでは関西職業能力開発促進センターで行っているセミナーの内容を紹介します。はじめに使用環境を紹介した後、各セミナーの内容を簡単に紹介します。セミナーにはJavaプログラミング導入技法、Javaプログラミング実践技法、Javaプログラミン

グ応用 (JavaBeans 編), Java プログラミング応用 (Swing+JDBC 編), Java Servlet があります。セミナーは体系化されており, 順に受けていくとエントリーから応用までできることとなります。

本内容では著者が担当していないJava Servlet以外の内容を紹介します。

### 3.1 使用環境

セミナーで実施している使用環境について紹介すると, OSはWindowsNT4.0 Workstationです。JDKについてはSun Microsystems社から無料配布されているJDKを使っています。バージョンについては1年目はJDK1.1.6, 2年目はJDK1.2.2, 3年目はJDK1.2.2\_001でした。統合開発環境としてインプライズ社のJBuilderやシマンテック社のVisual Cafeなどありますが使っていません。ちなみにJBuilderのFoundation版やIBMのVisual Age for JavaのEntry版(英語版)は無償となっており, 雑誌やホームページから取得できるようになっています。

一方, 作業環境ですが, エディタについてはVisual C++を使い, コンパイルと実行についてはJavaプログラミング導入技法だけDOSプロンプトで行い, 残りのものはVisual C++の中で行っています。Visual C++に簡単な細工をすると, コンパイルやアプリケーションの実行やAppletviewerの実行などをVisual C++の統合開発環境の中で行うことができます。

### 3.2 Javaプログラミング導入技法

Javaプログラミングの最初のセミナーとしてJavaプログラミング導入技法を用意しています。本セミナーの対象者としては基本的にはC言語の基本的文法を知っていることを前提としています。これはJavaがオブジェクト指向言語なので初心者には難しいと考えました。それで, 基本的な文法が似ているC言語を知っていることを前提としています。そして, Javaプログラミング導入技法ではテキストと市販の本<sup>4)</sup>を使っています。基本的にはテキストをベースに進め, 補足的な紹介として市販の

本を使っています。テキストは全部で13章あります。以下に章のタイトルを列挙します。

- 第1章 Java
- 第2章 Javaプログラミングの概略
- 第3章 プログラムの基本要素
- 第4章 クラスとオブジェクト
- 第5章 配列
- 第6章 文字列
- 第7章 static修飾子
- 第8章 データ隠蔽
- 第9章 パッケージ
- 第10章 例外処理
- 第11章 ストリーム
- 第12章 スレッド
- 第13章 AWT入門

そして, これら章を通じて以下の項目を紹介しています。

- (1) Javaの特徴
- (2) クラスとオブジェクト
- (3) プログラムの基本要素
- (4) 例外処理
- (5) 入出力
- (6) スレッド

Javaプログラミング導入技法ではJavaアプリケーションでこれらのことを行っています。内容ではJavaアプリケーションコードの作成方法とJavaの文法をはじめに紹介します。その後でクラスの紹介をします。クラスの紹介に入るとオブジェクト指向言語の特徴が現れて, はじめてオブジェクト指向に触れる方には理解するまでに時間を要します。その他, ライブラリとしてのパッケージを作成したり, 例外処理やスレッドプログラミング, ファイル入出力などを行います。スレッドプログラムは他の言語で行うことを考えれば非常に簡単にできます。加えて, スレッドはJDK1.2.Xで大きく変更されています。これまではスレッドクラスのstopメソッドなどを使っていましたが, 問題があって使うことを推奨されなくなりました。JDKのドキュメントに掲載されているのでそちらを参照してください。また, 終わりには次のJavaプログラミング実践技法につ

なげるために、簡単なGUIプログラミングを時間があれば行っています。本セミナーではJavaの基本要素を紹介しているので、本セミナーを生かして展開を考えれば、標準パッケージの紹介やアプリケーション開発を加えていけばよいと思います。

### 3.3 Java プログラミング実践技法

Javaプログラミング導入技法でJavaの基本要素を理解した後に、Javaプログラミング実践技法を用意しています。本セミナーではメインがアプレットとGUIになっています。対象者としてはJavaプログラミング導入技法を受講している、または同等な知識をもっていることを前提としています。そして、Javaプログラミング実践ではテキストと市販の本<sup>5)</sup>を使っています。基本的にはテキストで授業を進め補足的な紹介で市販の本を使っています。テキストは全部で5章あります。以下に章のタイトルを列挙します。

第1章 アプレット

第2章 イベント処理

第3章 GUI

第4章 イメージ表示のテクニック

第5章 GUIアプリケーション

これら章を通じて以下の項目を紹介しています。

- (1) アプレット
- (2) GUI
- (3) イベント処理
- (4) アニメーション
- (5) GUIアプリケーション

本セミナーではブラウザ上とOS上の2種類の動作環境を使います。ブラウザにはAppletviewerやIE(Internet Explorer)やNN(Netscape Navigator)を用意しており各自で試してもらっています。現在使っているのはIE5.0, NN4.6です。これらはJDK1.1.xをサポートしているのでJDK1.2.xのコードを実行できないことがあります。結果として一番動作するのがAppletviewerとなってしまいます。セミナー受講者からすれば使われているブラウザで試したいということになりますが、現実にはブラウザの対応が遅れています。本セミナーでは、はじめ

にアプレットの作成方法およびブラウザ上での実行方法を紹介します。その後、細部に入ってグラフィックスやイベント処理およびGUI部品の紹介を行っています。アプレットの作成はさほど難しくありません。アプレットの基本となるクラスがあらかじめ用意されているのでそれを継承して作成するだけです。大きな内容としてはイベント処理があります。イベント処理はJDK1.0からJDK1.1にバージョンアップするときに大幅に変更されています。JDK1.0では1つのクラスの中でイベント発生とイベント処理が行われていました。これがJDK1.1からはイベント発生側とイベント処理側にクラスを分離することが可能になりました。イベント発生側のことをイベントソース、イベント処理側のことをイベントリスナと言います。また、このイベント処理モデルは委託モデルとかデレゲーションモデルと呼ばれています。イベント処理を理解できればイベント駆動型のプログラミングを行えるようになります。

### 3.4 Java プログラミング応用(JavaBeans 編)

Javaプログラミング導入技法とJavaプログラミング実践技法で一通りのJavaプログラムの紹介をしたことになるので、後は個別技術の紹介になります。Javaではコントロールを開発する内容としてJavaBeansがあります。JavaBeansはプログラムの再利用を目指したJavaにおけるソフトウェア部品開発のフレームワークです。概念的にはコンポーネントウェアという言葉でくくられており、同じような位置づけにMicrosoft社のActiveXコントロールがあります。Javaプログラミング応用(JavaBeans 編)ではテキストと市販の本<sup>6)</sup>を使っています。基本的にはテキストで授業を進め、補足的な紹介を本で行っています。テキストは例題形式にしており以下の通り全部で11例題を用意しています。

例題1 JavaBeansを理解する

例題2 BeanBoxでJavaBeansを使う

例題3 BeanをJavaコードで使用する

例題4 Beanをアプレットで使う

例題5 Beanを作る

例題6 イベント作成

例題 7 情報公開 ( BeanInfo クラス )

例題 8 プロパティ値の変更

例題 9 プロパティエディタ

例題 10 カスタマイザ

例題 11 シリアライゼーション

すべてを通して一通りの内容が備わった Bean ができ上がるように紹介しています。作成した Bean は BDK ( JavaBeans Development Kit ) に入っている BeanBox アプリケーションでテストを行います。BDK については Bean 用の開発キットで、中にはビルダーツールのデモもしくは Bean のテストコンテナとして使われる BeanBox と Bean のサンプルおよびドキュメントが入っています。BeanBox はビルダーツールの参考例として取り上げられており、市販されているビルダーツールはこの BeanBox を参考に開発されているとのことです。Bean 開発用の API は Java Core API に含まれているので JDK の中に入っています。ですから、BDK は Bean 開発に必須ということではありません。

本セミナーの内容では、はじめに JavaBeans の概要を紹介します。概要の紹介では Bean であることの必要条件を説明し、合わせて Bean の利用方法を紹介します。概要を理解した後は Bean 開発方法を紹介します。Bean 開発は決められた形があるのでその手順に従ってさえいれば作成することができます。理解が困難なところは用意したクラスやメソッドなどがどこで呼び出されているか見えないところです。プログラマーが用意したプログラムを他のプログラムの中でコールするのであればプログラムをたどっていただけで理解できますが、そうではないところがあります。ですが、とても理解できないということでもありません。工夫をすれば変化を見ることができると思います。

Bean の作成は形式的な内容が強いので Bean 開発のアイデアが勝負となります。つまり、何を作るかです。コンポーネント開発はとにかくアイデアが決め手となります。Active X コントロール開発にも同じことが言えます。実際に練習問題を作るときもどんな Bean を用意すればわかりやすいか考えさせられました。

### 3.5 Java プログラミング応用

( Swing+JDBC )

Java プログラミング応用 ( Swing+JDBC ) には 2 つの内容が入っています。Swing と JDBC です。Swing は JFC ( Java Foundation Classes ) の 1 つで GUI コンポーネント群のことを指します。もともと Java には GUI 部品として標準的に AWT ( Abstract Window Toolkit ) がありましたが、部品数が少ないことやできることが貧弱であると指摘されていました。それをより改良したものとして Swing が開発されました。Swing の特徴としては部品の種類を増やしたことに加えて Light-Weight コンポーネントであることがあげられます。Light-Weight コンポーネントとはピアクラスを使っていないコンポーネントのことを言います。これまでの AWT はピアというクラスを見えないところで使っていました。ピアとはシステム差分を吸収しているクラスです。プログラマーはシステム差分のことを考えなくプログラミングを行えますが、中ではブラックボックス化された形でピアが働いています。要するに OS が代わっても内容が同じであれば記述するプログラムには相違が無いということです。ちなみにピアの部分にプログラマーは踏み込めないことになっています。ピアを使っている AWT では結果としてシステムのライブラリを使っています。例えば、Windows で考えると Win32 が AWT から利用されており、Visual Basic で使われている部品と同じものであったりします。一方、Swing はすべて Java コードで記述された部品でシステムのライブラリを使いません。ちなみにピアを使っているコンポーネントは Heavy-Weight コンポーネントと呼ばれています。

次に JDBC ですが、JDBC は Java コードからデータベースに接続する規格です。実際には Java コードから用意されている JDBC ドライバーソフトウェアを使ってデータベースに接続します。これと同じような規格に Microsoft 社が提唱した ODBC ( Open DataBase Connectivity ) があります。ODBC もドライバソフトウェアを用意してどのデータベースでも接続できるようにする規格です。Windows で

は標準的にODBCのセットアップ環境が備わっています。

JDBCのドライバーソフトウェアには4種類のタイプがあります。JDBCのホームページを見るとデータベースベンダをはじめさまざまな会社が各タイプのドライバーソフトウェアを提供しています。そして、これらドライバーソフトウェアを使ってアプリケーションやアプレットからデータベースに接続します。

本セミナーではSwingとJDBCをそれぞれ個別に紹介しています。練習問題でこれら2つの内容を含めた形で提示しています。本セミナーではテキストと市販の本<sup>7)</sup>を使っています。基本的にはテキストで授業を進め、補足的な紹介を本で行っています。以下のとおり全部で11例題を用意しています。

- 例題1 Swingを理解する
- 例題2 フレーム作成
- 例題3 プラグイン可能なルック&フィール
- 例題4 モデルを使う
- 例題5 JTableを使う
- 例題6 JTreeを使う
- 例題7 JTreeの項目を変更する
- 例題8 JDBCを理解する
- 例題9 日本語に対応する (SJISからUnicode)
- 例題10 日本語に対応する(UnicodeからSJIS)
- 例題11 PreparedStatementクラスの使用

Swingについては部品が収まっているパッケージの紹介と、プラグイン可能なルック&フィールの説明と、Swingが採用している部品モデルのUIデリゲートについて紹介しています。合わせていくつかのSwing部品を使ったプログラミングを紹介しています。SwingのパッケージはJDK1.1.xのころはJava Core APIに入っていなかったためパッケージ名なども別で用意されていました。JDK1.2.xからはJava Core APIに入ったためパッケージ名も標準的な名前に変更されました。プラグイン可能なルック&フィールというのは見栄えを切り替える機構と見栄えを提供するクラスのことを言います。Swingでピアを使わないようにしたので見栄えに問題がありました。ピアを使っているAWTはシステム固有

のライブラリを使っており、見栄えなどはライブラリに依存した形になっていたため考える必要はありませんでした。しかし、Swingではピアを使わないので見栄えの問題があり、これをプラグイン可能なルック&フィール(pluggable look and feel : plaf)を用意して解決しました。また、ピアを使わない結果としてコンポーネントをすべて自分で作成できることとなります。AWTなどを使った場合は必ずピアが入っているため見栄えなど作ることは不可能でした。中でも紹介しましたが、ピアの部分にはプログラマーは踏み込むことができませんでした。

続いてUIデリゲートですが、これは部品のモデルを指しています。Swingの部品はこのモデルを使って作成されています。これはもともとSmalltalkで採用されていたMVC(Model-View-Controller)モデルをベースに考案されました。ちなみにMVCモデルはMFC(Microsoft Foundation Class Libraries)のドキュメント・ビューアーキテクチャの基礎にもなっています。UIデリゲートはViewとControllerを一緒にしたものとModelの2つで構成されます。本セミナーではSwing部品を使ってこのUIデリゲートを紹介しています。UIデリゲートモデルはかなり複雑なので簡単な部品には適しませんが、大きな部品を作るときには整理がついてプログラミングの労力が改善されるということです。また、Swing部品についてはJFrameやJCombo, JList, JTable, JTreeなどを使っています。

JDBCについてはJDBCの概要やJDBCドライバーの種類を紹介をします。その後、JDBC APIを使ってデータベースに接続する方法や文字コードの不具合、有用なクラスなどを紹介しています。セミナーの中ではJava Core APIと一緒に入っているJDBC-ODBC Bridgeドライバを使ってデータベースに接続しています。このドライバはJDBCとODBCのマッピングを行います。データベースと直接結びついているのはODBCドライバということになるのでODBCの設定も必要です。接続するデータベースにはMicrosoft社のAccessを使っています。文字コードの不具合については書籍<sup>8)9)</sup>などでも紹介されています。Javaでは文字コードとして

Unicodeが使われていますが、ここで使っている Accessには文字コードとしてSJISが使われています。本来であればJDBCドライバが異なる文字コードの処理を吸収することになってはいますが、現状ではここがうまくいっていないので便宜的に文字コード変換をプログラムの中に入れてあります。データベース接続のプログラミングについてはJDBCドライバでデータベースの差分を吸収しますので、どのデータベースでも同じようなプログラミングになります。これがJDBCのメリットです。また、接続するために必要なクラスやメソッドも標準パッケージで提供されているので難しくはありません。データベースからデータを取り出すプログラムは手続き的になります。

### 3.6 セミナー実施状況

Javaに関するセミナーは平成10年から開始しています。表1に平成10年からの応募状況を示します。コース数は年々確実に整備され充実させています。また、応募人数については定員オーバーの人数は含まれていません。各年度の総数を比較すると、平成10年度の定員総数が96名のところ応募総数が87名、平成11年では定員総数が107名のところ応募総数が100名、平成12年では定員総数が166名のところ応募総数160名となっています(図4参照)。定員の充足率から考えると平成10年が91%、平成11年度が93%、平成12年度が96%と年々向上しており、Javaに対する関心の高さがうかがえます。また傾向としてJavaプログラミング導入技法から

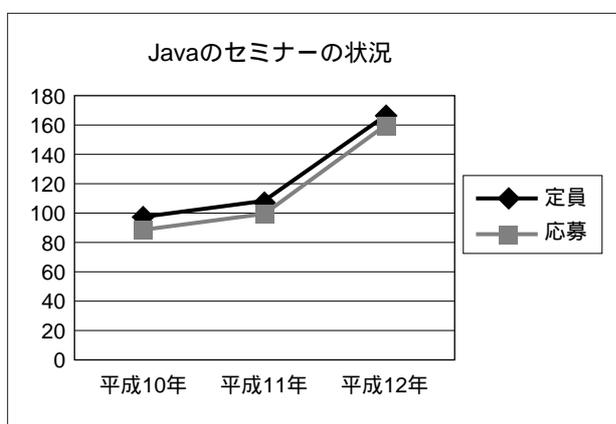


図4 定員と応募の総数

入ってJavaプログラミング応用まで段階的な受講の傾向があり、こちらが当初考えた形で段階的にセミナー実施できる方向になってきています。

### 4. おわりに

Javaに関しては未だに盛んに開発が行われ新しい技術が生み出されています。開発内容も多岐にわたり広がりを見せています。また、Java対応のOSなどの開発により機器に対する広がりも見せています。企業にとって商品価値のある題材として盛んに研究・開発が行われています。現在では機構が作成している標準モデルカリキュラムにもJavaプログラミングに関するモデルカリキュラムがいくつか掲載されています。需要が合えば能力開発セミナーとしてこのJavaプログラミングをやってみてはどうでしょうか。

問題として市販の本が少ないことや統合開発環境のツールが他の言語に比べて充実していないことがあります。市販の本に関しては副読本の選定にかなり苦労しました。Javaのバージョンアップが激しいので本が追いついていかないことがあります。また、統合開発環境のツールに関しては、使うのは簡単ですが、自動的に作成されたコードを初心者が理解するにはかなり難があると思えます。そして、問題が発生したときに対処できないような気がします。GUIを操作してフォーム等を作成してもすべてソースコードが生成されます。ですから、ソースコードが読めないと問題が発生してもどこがおかしいのかわかりません。紹介したJavaプログラミング導入技法やJavaプログラミング実践技法を経験した後に、統合開発環境のツールを使うのが適切かと思われず。

一方、日経BP社調査部によるJava利用実態調査<sup>10)</sup>ではJavaユーザが利用している開発ツールとしてはJDKそのままが4割でトップ、JBuilderがこれに肉薄したという結果でした。これを考えればJDKそのままの形でセミナーを開催してもさほど問題ではないと思われず。JDKは無料なので導入については統合開発環境を揃えることを考えれば

表1 セミナー実施状況

年度	セミナー名	定員総数(人)	定員(人)	応募総数(人)	応募人数(人)
平成10年	Java プログラミング導入技法	48	18	44	18
			18		14
			18		18
	Java プログラミング実践技法	48	18	43	15
			18		10
			18		18
平成11年	Java プログラミング導入技法	31	16	31	16
			15		15
	Java プログラミング実践技法	46	15	41	10
			15		15
			16		16
	Java プログラミング応用	30	15	28	14
15			14		
平成12年	Java プログラミング導入技法	46	15	46	15
			15		15
			16		16
	Java プログラミング実践技法	30	15	27	12
			15		15
	Java プログラミング応用 (Java Beans)	30	15	27	15
			15		12
	Java プログラミング応用 (Java Swing+JDBC)	30	15	30	15
			15		15
	Java Servlet	30	10	20	10
10			10		

非常に楽です。

能力開発セミナーの拡充を考えると、関西職業能力開発促進センターではJava言語でのネットワークプログラミングを行っていません。Java言語を使ってネットワークプログラムを開発するのは他の言語と比較すると楽に行えます。標準パッケージの中にネットワークプログラム開発用クラスがあらかじめ用意されているので、それらを利用すればかなり楽にプログラム開発を行えます。Windowsでネットワークプログラムを開発することを考えれば、基本的にWinsockのプログラムを開発することになります。Winsockのプログラム開発ではVisual C++の開発環境があり、Winsock用のライブラリが用意されています。Visual C++でWinsockのプログラムを作成することをJava言語でネットワークプログラムを作成することと比較すると、後者のほうが比較的容易に行うことができるように思えます。ただし、書籍の充実から考えるとJava言語での

ネットワーク開発の書籍はいくつかありますが<sup>11)12)</sup>、Visual C++のほうが豊富になっています。

Java言語の良さを考えるときにいつも比較対象としているのがVisual C++です。Java言語とVisual C++は共にアプリケーションを開発することができます。特徴から言えば、Javaは標準パッケージが用意されていて、それらをうまく使っていけばかなりの内容を作ることができます。標準パッケージは使う側からすればブラックボックス化されていますが、そこだけが不明確です。Visual C++のほうはアプリケーション開発には基本的にMFCを使います。MFCは巨大なライブラリで使い方を覚えるだけでも膨大になります。Javaの標準パッケージとは比較になりません。また、MFCを使うときはライブラリと合わせてマクロを使います。マクロは展開されればプログラムと同じですが、展開されたあとのプログラムを見るとぞっとすることがあります。例えば、メッセージ処理のマッピングを

しているマクロのBEGIN\_MESSAGE\_MAPやEND\_MESSAGE\_MAPです。コンパイルするときにはマクロの部分も影響するのでブラックボックス化されているのがMFCとマクロの2つになります。MFCの場合は隠れている部分が多いので問題が起きたときかなりの知識と経験が必要になります。ちなみに、Javaの標準パッケージとMFCは共にソースは公開されているので調べれば中身はわかりませんが、MFCはプログラムの行数が尋常ではありません。

また、コンポーネント開発に目を向けるとJavaにはJavaBeans、Visual C++にはActiveXコントロールがあります。これらの開発を比較するとBeanを開発するほうがかなり楽だと思われまが、プログラムの大きさや覚える内容はJavaBeansの方が少なくて済みます。Visual C++ではMFCを使ってActiveXコントロールを開発していました。結局、MFCを使っているのでアプリケーション開発と同じように覚えることがたくさんあります。最近ではATL(Active Template Library)を使って開発することもできるようになっており、かなり改善はされています。それでもBean開発に比べれば雲泥の差があると思います。

#### 参考文献

- 1) 堀内泰輔訳：Javaプログラミング技法，ピアソン，1997.10.31.
- 2) 武舎広幸ほか訳：Java言語入門，ピアソン，1997.6.1.
- 3) 翔泳社：DDJ JAPAN，1997.1月号，p.30～p.33.
- 4) 結城浩：Java言語プログラミングレッスン(下)，ソフトバンクパブリッシング，2000.2.9.
- 5) 小嶋隆一訳：JavaプログラミングVol.2，ピアソン，1999.12.25.
- 6) 坂本衛：JavaBeansプログラミング入門，オーム社，1997.11.20.
- 7) 福井眞吾ほか訳：続・Java言語入門，ピアソン，2000.10.25.
- 8) 日本サン・マイクロシステムズ編：JDBCサイエンス社，1998.7.10.
- 9) 菊田英明：実践JDBC，オーム社，1998.5.12.
- 10) URL：http://res.nikkeibp.co.jp/r-marketing/rep012.html
- 11) 戸松豊和監訳：JAVAネットワークプログラミング，オーム社，1997.7.
- 12) 小高知宏：基礎からわかるTCP/IP Javaネットワークプログラミング，オーム社，1999.11.30.
- 13) 光澤敦他訳：プログラミング言語Java，星雲社，1997.9.25.
- 14) 岩谷宏訳：速習JavaSwingプログラミング，ソフトバンクパブリッシング，1999.8.30.
- 15) 日本サン・マイクロシステムズ編：JavaBeans，サイエンス社，1997.12.25.
- 16) 武藤健志監修：独習Java，翔泳社，1999.8.30.
- 17) 戸松豊和：JavaBeansマスターブック，翔泳社，1998.12.10.
- 18) 大村忠史：SwingによるJavaGUIプログラミング，カットシステム，1998.6.10.
- 19) 大村忠史：SwingによるJavaGUIプログラミング( )，カットシステム，1998.11.10.
- 20) 高橋弘行：JAVAプログラミング講座，アスキー出版局，1998.4.21.
- 21) 萩本順三：やさしいJavaプログラミング改訂版，エアアイ出版，1998.11.13.
- 22) 株式会社インフォビジョン：JDBCによるデータベースアクセス，アスキー，1998.9.1.
- 23) 日本サン・マイクロシステムズ編：インターフェイス総説，サイエンス社，1997.4.25.
- 24) 玉井浩訳：MFCインターナル，星雲社，1997.5.12.
- 25) 株式会社クイック訳：ActiveX Controls Inside Out 改訂新版，アスキー出版局，1997.11.1.
- 26) 豊田孝訳：ATLプログラミング，ソフトバンク，1998.8.1.
- 27) 田中正造監訳：ATL COMプログラミング，翔泳社，1999.1.25.
- 28) 掌田津耶乃：VisualCafe ver.3で学ぶJavaプログラミング入門，技術評論社，1999.4.25.
- 29) 藤井等：JBuilder オフィシャルコースウェア 入門編，アスキー，1998.4.21.
- 30) 藤井等：JBuilder オフィシャルコースウェア 応用編，アスキー，1998.4.21.
- 31) 江村豊監修：Winsock2.0プログラミング，ソフトバンク，1998.8.26.