経営論集 第11巻第１号 2001年　129～151頁

Java 言語プログラミング教育教材の考察

――BASIC 言語から Java 言語へ――

海老澤　信一・矢野口　　聡

はじめに

コンピュータが出現してからほぼ半世紀が経過した。その間，コンピュータは軍事を中心とした特殊な利用から始まり，大企業や大組織を中心とした大型コンピュータの応用へと移り，コンピュータ同士を接続した分散処理型形態を経由して，パソコンを活用した個人や SOHO の利用すなわち EUC（End User Computing）へと急速に拡大した。いわば点から線，線から面，面から立体への拡大である。コンピュータの相互接続は数年という短い期間で爆発的な勢いで世界に普及し，いまやインターネットが世界を変えるまでの影響を与えているのは周知の事実である。

ハードウェアと共にソフトウェアの発達も大きく変化してきた。現代社会のコンピュータ技術とその利用を支えるため，1980年代以降大学・短大・専門学校に情報教育が積極的に取り入れられ，教育界から沢山の人材が社会に供給された。この間，ソフトウェアを中心とした情報教育内容は大きく変化している。1980年代は COBOL，FORTRAN，BASIC 等のプログラミング言語教育が主流を占め，1990年代前半はワープロ，表計算，データベース等のアプリケーション利用教育がそれに続いた。1990年代後半はグラフィックス，インターネット，ホームページ等の利用や作成教育が開花した。21世紀の情報化社会では，プロダクツ（各種製品群），ネットワーク（高速大容量通信），コンテンツ（制作された内容）がバランスを保って発達することが重要である。とりわけ我々が直接触れるコンテンツの善し悪しが，そのシステム自体が利用されるか否かを左右する。その意味で Web プログラミング技術は今後欠かせない技術となり，高等教育機関における情報教育にも広く取り入れられるべき技術となろう。そのような観点から，筆者らは本学部のような文科系学部で，Web プログラミング技術教育の一環として Java 言語をどのように取り入れ教育すべきかを研究している。本稿はその教材作成の成果の一部である。

1　Java 言語の生い立ち

1－1　Java 言語の誕生

Java 言語は，UNIX コンピュータの開発メーカーである米 SunMicrosystems 社が1995年に発表したプログラミング言語である。源流はその５年前に Sun 社が始めた「グリーンプロジェ

クト」にある。このプロジェクトの目的は，家電製品の制御などに使う組込用言語「Oak」の開発であった。家電製品に搭載されるプロセッサは頻繁に変更されるため，その都度プログラムを書き換えなくても済むような言語の開発が求められていたわけである。プラットフォーム（機種やOS）を選ばない「Oak」の技術は組込用言語として評価されただけではなく，ワークステーションやパーソナルコンピュータ向けの大規模ソフトウェアの開発者達にも評価され，本格的なプログラミング言語「Java」として登場した。

その当時の米国はインターネット黎明期であった。1993年から1994年にかけてMosaicやNetscape Navigatorといった Webブラウザが登場したことにより，インターネットの爆発的な広がりを予測したSun社は，まずJava言語を Webブラウザの組み込みソフトウェアを開発するための言語として普及させようとした。1996年には Netscape Navigator2.0に Java言語が正式にサポートされ，まもなく Internet Explorer3.0にもサポートされたことで，Java言語は Webブラウザ用サポート言語として一躍メジャーの舞台に上がることになった。その要因は，Sun社が Java開発環境の JDK1.0（Java Development Kit 1.0）および実行環境の HotJavaを無償配布したことであろう。JDKはその後1.2，1.3と版を重ねより高速で安定したことで，本格的なプログラミング言語としても有用であることが広く世間に認知され普及することに成功した。

1－2　C言語から Java言語へ

Java言語の利点はプラットフォームを選ばないという点だけではない。他のメジャーな開発言語であるC言語と比較した利点は次のようである。すなわちC言語ではプログラムの分割や結合の作業が難しいために，長いプログラムになると開発者本人でさえコード全体の把握が困難になり，デバッグに多大な時間がかかるという欠点がある。これを解決すべく登場したのがオブジェクト指向を取り入れたC++言語であった。オブジェクト指向の利点は，「クラス」と呼ばれるオブジェクトの原型を用意しておき，必要な機能はクラスを使ってそのシーンに合ったオブジェクトを生成することができる点にある。これにより，本来ならば何百行にわたるコードを書く必要があった Windows表示プログラムなども，開発元が提供するクラスライブラリを利用すれば，１行の記述で済んでしまう。また，ユーザは開発元が提供しているこのクラス（スーパークラス）に独自の機能を追加して新たなクラス（サブクラス）を作ることもできる。これは「継承」と呼ばれるプロセスであるが，これらの機能を活かせば多人数での共同開発が容易となる。このことは開発期間を短縮させるだけでなく，プログラムそのものの質を高める効果もある。開発者それぞれが作ったクラスをネットワーク上で共有させ，多人数の目に通すことでブラッシュアップされより質の高いものとなっていく。また，クラスは完全に独立しているため，開発者それぞれが得意とする部分を集中してコーディングすることができる。つまり，現実的なコラボレーションによる開発環境が整備されたわけである。また，オブジェクト指向プログラミングにおける開発効率の向上は，開発元がどれだけ多くのクラスライブラ

リを提供するかにもかかっている。実際，多くの開発ベンダーがMicrosoft社のVisualC++言語を採用した理由は，付属のMFC（Microsoft Foundation Class）に，非常に多くのクラスが用意されており，さまざまな機能を容易に実現することができたことにあった。

しかし，Webや携帯端末の技術の発展により，ソフトウェア業界の市場に変化が起きてきた。すなわち，アプリケーションの動作環境が，スタンドアローンコンピュータやLANで接続されたシステムからインターネットへと移行してきたのだ。Web上で動作するアプリケーションは，Webブラウジング機能を持つ端末であればプラットフォームを選ばず，モバイル機器との相性も良い。また，eコマースなどビジネス支援ツールとして積極的にWeb技術を取り入れようとする企業の需要も高まってきた。この状況はC++言語によるプログラム開発の限界を露呈することとなった。

まず，LANと比べて極端に回線帯域の狭いインターネット上にプログラムを配信するにはできるだけサイズを小さくする必要がある。これに対し前述のVisual C++言語で生成した実行可能形式のプログラムは，MFCやDLLといった付随プログラムが含まれるため，最低でも５MB以上になってしまう。ブロードバンドが主流になりつつある現在でも，インターネットは数MBものプログラムを頻繁に回線上に流せるような状況ではない。また，C++言語がプラットフォームに依存しないと唱えているのはソースレベルでの話であり，受信側が実行形式のファイルを実行できない場合も多い。次にセキュリティの問題は，ビジネスに耐えうるWebアプリケーション開発には重要な要素である。C++言語にはプログラムの処理時に不正な動作を検知するような機能を持たないため，本来の機能に加えてこのセキュリティ機能を追加コーディングする必要がある。これも開発側にとって大きな負担となる。

１－３　Java言語の形態

Java言語はこれらの問題を解決するものである。Java言語では，JVM（Java Virtual Machine）と呼ばれるJava仮想マシンを用意し，プログラムの実行はこのマシン上で行うという思想を取り入れている。JVMが実装されていれば，プラットフォームの違いは関係なくなる。つまり，ソースプログラムはJVMが解釈可能なバイトコードにコンパイルされ，実際の実行はJVMの内部で１ステップずつ機械語に変換して処理されていく。コンパイルという処理はあくまでもJVM上での処理であり，内部的にはインタプリタとして動作している。インタプリタは動作速度に難があるという欠点もあるが，Java言語は同時に複数の処理が行えるマルチスレッド方式を採用しているため，見かけ上ストレスを感じることは少ない。インタプリタ自体もバージョンを重ねるごとに改善され，より高速な処理が可能となっている。もっとも，最近のコンピュータの処理能力からすれば，コンパイラ言語とインタプリタ言語の差はないと言っても過言ではないであろう。また，JVMにはプログラム実行中にその動作を監視したり，Javaコードに署名を付けて，プログラムの改変を防止したりといったセキュリティ機能を持っている。これは従来の言語とは大きく異なる点である。Java言語は，このようにインターネッ

トでの利用を考慮し予め必要な機能を JVM に持たせることで，配信するプログラムのサイズを小さくすることができたわけである。

現在，Java 言語の開発・実行環境は，Windows，MacOS，Solaris，HP-UX，UX/4800，Linux, FreeBSD などのプラットフォームに対応しているが, JavaChip の開発によりコンピュータだけでなく，i-mode，NC（ネットワークコンピュータ），PDA（携帯情報端末）や家電製品での動作も可能となっている。作成したプログラムをサイト（Web サーバ）側に置き，ユーザ側の端末にダウンロードして Web ブラウザやビューア上で実行する形態が最も一般的である。このような Java プログラムはアプレットと呼ばれているが，Java 言語が登場した頃はアプレットの読込速度が遅い，Web ブラウザの動作が不安定になる，などの問題が多かった。しかし，その後 Java HotSpot Performance Engine の開発によって，安定性と処理速度が飛躍的に向上している。また，開発効率を上げるための工夫として JavaBean と呼ばれる技術も考え出されている。これは，他の Java プログラムから簡単に呼び出して使用可能なコンポーネントと呼ばれるプログラム（Bean）を提供するもので，JavaAPI（標準で用意されている Java のライブラリ）よりもさらに使いやすい仕組みである。

一方，スタンドアロン上で動作させる形態のプログラムをアプリケーションと呼んでいて，他の言語と同様に通常のプログラミングを行うことができる。最近は，Web サーバ上で動作させるサーバサイドプログラミングでの用途が主流となっており，データベースサーバと連携させてグループウェアやショッピングサイトなどの Web アプリケーションの開発に使われるようになってきた。このような形態のプログラムを特にサーブレットと呼んでいるが，Java 言語はイントラネット開発環境の本命と言える。

Java 言語を Web サーバ上で動作させるためには，JSP（Java Server Pages）と呼ばれる Microsoft 社の ASP（Active Server Pages）に相当する仕組みを必要とする。JSP では HTML ソース内に直接 Java 言語のコードを記述し，Web サーバがその記述を解釈して動的に HTML ソースを生成・送信する仕組みになっている。そのため，Web サーバにこの JSP エンジンを組み込んでおく必要があり，いくつかのエンジンが用意されている。Tomcat，JRun，WebLogic，WebSphere，iPlanet などが出回っている。また，JSP にも EJB（Enterprise Java Bean）と呼ばれる Bean が存在するので, 高度な Web システムを効率よく開発することができる。その他，最近注目されているのが i-mode 向けアプリケーション開発である。i-mode で動作する i アプリは Java プログラムであり, 組み込みプログラム言語としての能力を充分に発揮できる分野であるといえる。i アプリの動作は，家電製品などにも組み込まれている KVM（K Virtual Machine）が採用されている。

現在, Java 開発環境の核となる JDK には 3 つのエディションが存在する。Standard Edition（Java2 SE）は最も基本的なキットで，個人利用や言語学習の場合にはこれを利用することが多い。Enterprise Edition（Java2 EE）は Standard Edition に拡張キットを追加したもので，JSP を使った Web システムの構築に適している。また，i アプリ等の組み込みプログラム開発

にはMicro Edition（Java2 ME）が用いられているが，これもStandard Editionの拡張版である。Javaの開発環境は現在JDK以外にもいくつか存在しており，Borland社の「J Builder」，IBMの「Visual Age for Java」，Symantic社の「Visual Cafe」などのいわゆる統合開発環境（IDE）がよく使われているようである。これらの開発環境は，Visual BasicのようにGUI設計やデバッグ，コーディングなどが非常に容易に行え，生産効率を飛躍的に向上させることができる。なお，Sun社自身も「Forte for Java」をリリースしている。

2　Java言語プログラミング教育の導入

2－1　導入の背景

これまで様々な業界において多くの企業がeビジネスの武器としてWebサイトの構築に力を入れてきた。しかし，これだけ多くのサイトが出現している今，単にWebサイトを立ち上げるだけでは効果の薄いものとなってきている。Web上に多くのコンテンツを用意し，閲覧者にとって有用な情報を素早く提供することがポイントとなるであろう。そのためにはデータベースの構築が必要となるが，SOHOあるいは自社レベルでの構築では限界がある。今後は，既存の他サイトや他システムと連携させることで，量や質を補う方法が採られてくるであろう。例えば，普段様々なサイトへアクセスして必要な情報を収集し，レポートを作成するといった作業をWebサイトが代行してくれるシステムが出現することが予測される。そこにWebプログラミング技術は欠かせないものとなる。Java言語はこのような高度なシステムをエンドユーザが構築できる最も適した言語といえる。そのための基礎教育を経営学など文科系学部の教育に取り入れることは意味のあるものだと考えている。

1980年代，多くの文科系の教育機関で行われていた情報教育は高級言語を利用したプログラミング教育であった。筆者（海老澤）もCOBOL言語プログラミング技術やBASIC言語を利用したアルゴリズム教育の書籍を出版した経緯がある。当時，BASIC言語が教育で利用された理由は，優れたアプリケーションソフトがまだ世の中に存在しなかったこと，その名が示すように理解しやすい言語なので初心者にも言語体系全体を見渡せること，また実行環境の構築や実行が手軽であったこと，初歩的なグラフィックスが比較的容易に扱えること等であった。

しかし1990年代，優れたワープロや表計算ソフトが創作され世の中に普及すると，もはやプログラミングを教える時代ではないという風潮のもと，情報教育はこれらのアプリケーションソフトを習得する教育へと急速にシフトしていった。その間，理工系学部や専門学校ではシステム開発言語のC言語を教育するところも多かったが，文科系学部ではC言語を取り入れるところが少なかったと思われる。その理由は，C言語はBASICやCOBOL言語などに比較して文字列が扱いにくいことや，学習者は初期の段階からポインタの概念を理解する必要があることや，BASIC言語だから教えることができた担当教員もC言語では充分な教育体系を確立できなかったのではないか等，問題点が多々あったと考えている。その後オブジェクト指向を取り入れたC++言語が普及したが，教育機関での取り組みは困難が付きまとった。特に文科系学部

では，CやC++言語プログラミング教育を導入するとしても現実には１つの選択科目として設置するしかないであろう。

2－2　導入の方法

筆者らが，Java 言語プログラミング教育を導入したいと考えた背景は既に述べたとおりである。しかし Java 言語の歴史はまだ浅く，充分なプログラミング教育体系が確立されている訳ではない。CやC++言語を取り入れたプログラミング教育体系が必ずしも成功するとは限らないように，Java 言語を導入しても成功するとは限らない。2，3 年前までは，Java 言語に関する市販の書籍類は技術者向けのものが多く，教育に適用できる書籍が少なかったが，最近は Java 言語に関する書籍類は数多く出版されいろいろな角度からのプログラミング技術論が展開されている。しかし学校教育に適用できる書籍は少ないので，筆者らは BASIC 言語教育が当時それなりの成功を収めた体験をもとに改めて教材を整理し，Java 言語教育に取り入れてみようと考えた。

Java 言語で難しいのは，クラスやオブジェクトの概念把握であろう。既製のライブラリやクラスを利用して，効率よく短時間でプログラムを組み上げる技術は，インターネットが発達して地域に分散した技術者がコラボレーション開発を行わなければならない現代のシステム開発技術では必須の技術である。しかし，文科系学部にとって学習者は初心者であることの方が多い。そこで，初心者が理解しやすかった BASIC 言語教材を整理し，Java 言語プログラミング教育にも適応できないかと考えた。具体的には，第１フェーズではクラスやオブジェクトの概念を教育することを避け，まず main メソッドの中で基本３構造（直線，分岐，繰返し）の基本的なアルゴリズム，配列やソート等の基礎となる技術を教育してプログラミングに慣れさせることを主眼とした教育方法の提案である。ある程度，Java 言語に慣れてきたところで，第２フェーズとしてクラスやオブジェクトの概念を把握させるという順序を踏む。充分に吟味した教育体系を用意しないと初心者にはプログラミング自体が敬遠されてしまう。Java 言語プログラミング教育体系の確立はなかなか難しい。以下に示すのは，筆者らが BASIC 言語プログラミング教育で培った教育体系と教材を改めて Java 言語に適用する試みである。

3　Java 言語教材試案（第 1 フェーズ）

3－1　基本 3 構造

オブジェクト指向プログラミングは，従来の手続き型プログラミングに比べて，「処理の流れ」そのものには注目しない。そのため，初心者は基本的なアルゴリズムを学習する機会が比較的少なくなる。特に演習時間が少ないときは，プログラムを作成する上で，最低限必要なロジック（例えば，直線・分岐・繰返しや比較・入れ替え等）を確認しないまま，オブジェクト指向の考え方から教育を始めるのは無理があろう。基本３構造を学習する教材は，クラス等の説明をできる限り簡単にし，メソッドの中でアルゴリズムのみに注目させるところから始める。例

題を提示して充分説明し，演習問題を与えて理解を深め，初期段階は時間をかけて教育すべきであろう。教材提示の順番は次のように行い，各段階の適当な場面で基礎的なアルゴリズムを理解させると良いと考えている。

⑴直線型の説明と演習
- ・出力
- ・入力
- ・入出力

⑵分岐型の説明と演習
- ・if else 型
- ・if 型
- ・ 3 分岐型

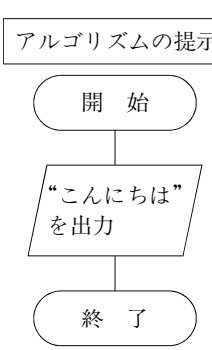
⑶繰返し型の説明と演習
- ・while 型　・for 型
- ・合計と平均
- ・最大値，最小値

⑷配列
- ・配列の宣言
- ・配列の初期化など

3－2　直線型の説明と演習

１）出力の演習（例題の提示）

出力の演習から開始するのはプログラミング教育の常道であり，まず画面への出力から始める教材の提示が求められよう。そこでこの段階ではアルゴリズムとプログラムの対応関係および System.out.println の説明に時間をかける。学習者には，場合によってはアルゴリズムを実際に考えさせてみることも必要であろう。ここではクラスとメソッドについては予備知識を与える程度で良い。プログラムの書き方として，空行，余白など学習者が理解しやすい書き方を提示し，説明ではそれを遵守すべきである。



Java 教材の提示

```
class Ex100 {

    public static void main (String args[]) {

        System.out.println (“こんにちは”);
    }
}
```

説明上強調すべき内容
- ①クラスについてのごく簡単な説明
- ②main メソッドについての説明
- ③画面出力の説明

２）出力の演習（題材の提示）

この段階では，①定数（文字と数字）と変数　②変数の型と識別子　③変数の宣言　④代入　⑤四則演算　などを必要に応じて説明する。但し，この時点の演習から大きくはみ出すような，

詳細な説明は避けるべきである。画面への出力演習としては，次のような題材を提示して定着を計る。題材の提示は単なる演習問題の羅列ではなく，その演習を通して知識の定着を計ったり，次のステップへの助走だったりと，必ず意味のある問題提示を心がけなければならない。例えば以下の題材では，題材 1 では System.out.println における文字列の扱い，題材 2 では変数と代入，題材 3 では代入と四則演算，題材 4 と 5 では四則演算と画面出力の応用等，提示する演習の意味と順序を厳選すべきである（以下，題材の提示とともに必要な場合はプログラムの主要部分のみ示す）。

題材 1　次のような画面表示を行う。

①“今日は良い天気です”

```
System.out.println（“今日は”＋“良い天気です”）；
```

②（計算結果の表示）

```
System.out.println（5＊6＋9/3－15）；
```

③“計算結果は XX です”

```
System.out.println（“計算結果は”＋（5＊6＋9/3－15）＋“です”）；
```

題材 2　変数 x に数値を代入し，その数値を画面に表示する。

```
int x；
x＝8；
System.out.println（“x は”＋ x ＋“です”）；
```

題材 3　変数 x と y にそれぞれ任意の数値を代入し，その合計を画面に表示する。

```
int x＝25；
int y＝56；
int z＝x＋y；
System.out.println（“z は”＋ z ＋“です”）；
```

題材 4　a＝20，b＝10を代入し，次の①と②の形で画面に表示する。

①

```
30      int a＝20；
10      int b＝10；
200     int wa＝a＋b；
2       int sa＝a－b；
        int se＝a ＊ b；
        int sh＝a/b；
        System.out.println(wa)；
        System.out.println(sa)；
        System.out.println(se)；
        System.out.println(sh)；
```

②

```
a＋b＝30
a－b＝10
a ＊ b＝200
a/b＝2

System.out.println(“a＋b＝”＋ wa)；
System.out.println(“a－b＝”＋ sa)；
System.out.println(“a ＊ b＝”＋ se)；
System.out.println(“a/b＝”＋ sh)；
```

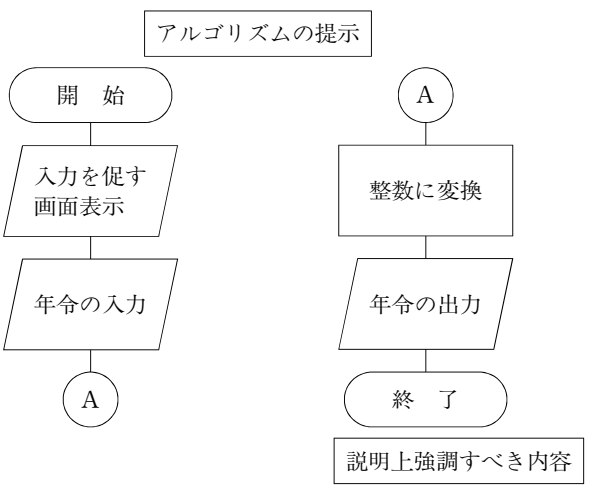
題材 5　単価 tanka と数量 suryo にそれぞれ100と10を代入し，売上げ uriage を画面に表示す

る。

```
int uriage＝tanka ＊ suryo；
System.out.println（“単価は　＝”＋ tanka ＋“円です”）；
System.out.println（“数量は　＝”＋ suryo ＋“個です”）；
System.out.println（“売上は　＝”＋ uriage ＋“円です”）；
```

3）入力の演習（例題の提示）

キーボードから数字を入力し，入力データに従ってプログラムを制御する教材を提示する。Java 言語では，キーボードからのデータ入力にはクラスライブラリを利用した次のような典型的なパターンがある。最初は入力命令はこのように書きなさいという説明で済ませ，入力データを使ったプログラム制御を中心に説明すべきであろう。




```
import java.io.＊；                                    ①クラスライブラリについての説明
    class Ex140 {
        public static void main(String args[]) throws IOException {　②入力エラーを標準で
        System.out.println（“あなたの年令を入れてください”）；　　扱うという説明
        BufferedReader reader＝new BufferedReader(new InputStreamReader(System.in))；
        String line ＝ reader.readLine()；　　③典型的な入力形式であるという説明
        int age＝Integer.parseInt(line)；　　④整数に変換するという説明
        System.out.println（“あなたの年令は　”＋ age ＋“歳　ですね！！”）；
        }　　⑤入力データを確認するという説明
    }
```

入力データのミスでプログラムがエラーとなるのを防ぐため，try{} catch{}が用意されているが，この段階で提示するとかえって学習者の理解を妨げることにもなりかねない。そこで，このプログラムでは入力ミス（必ず数字を入力させる）を起こさないようにして演習を徹底する。

4）入力の演習（題材の提示）

キーボードからの入力演習としては，次のような題材を提示して定着を計る。題材1はデータ入力と四則演算，題材3は入替えアルゴリズムの習得とワークエリアの意味の把握が演習のポイントである。

題材1　変数xとyにそれぞれ数値を入力し，その和を求めて結果を画面に表示する。

```
BufferedReader reader＝new BufferedReader(new InputStreamReader(System.in))；
System.out.println（“1つ目の数字を入れてください”）；　①変数Xの入力を促す
String line＝reader.readLine()；　②変数Xの入力
int x＝Integer.parseInt(line)；　③整数への変換
System.out.println（“2つ目の数字を入れてください”）；　④変数Yの入力を促す
line＝reader.readLine()；　⑤変数Yの入力
int y＝Integer.parseInt(line)；　⑥整数への変換
int z＝x＋y；
System.out.println（“合計は”＋z＋“です”）；
```

題材2　縦と横の長さを入力し，長方形の面積を求める。

題材3　入れ替えアルゴリズムである。xとyにそれぞれ数値を入力して表示し，次にxとyの内容をそっくり入れ替えて表示する。即ち入れ替え前と入れ替え後の数値を表示する。入れ替えには変数wを使用する（入力部分省略）。

```
System.out.println（“入れ替え前”）；　①入れ替え前の表示
System.out.println（“Xは　”＋x＋“　です”）；
System.out.println（“Yは　”＋y＋“　です”）；
int w＝x；　②入れ替え処理
x＝y；
y＝w；
System.out.println（“入れ替え後”）；　③入れ替え後の表示
System.out.println（“Xは　”＋x＋“　です”）；
System.out.println（“Yは　”＋y＋“　です”）；
```

5）入出力の演習（題材の提示）

キーボードからの入力と画面への出力を通して，次のような題材を提示して定着を計る。ここまでくると，学習者はかなり慣れてくるはずである。題材1から題材4まではデータ入力と四則演算と変数の扱い（例えば題材3では円周率3.14という数値にはdouble型を使う），題材5は入れ替えアルゴリズムの応用である。

題材1　底辺と高さを入力し，三角形の面積を求める。

double menseki＝teihen ＊ takasa/2；　　① double 型の説明
System.out.println（“底辺が　”＋ teihen）；
System.out.println（“高さが　”＋ takasa ＋“の時”）；
System.out.println（“三角形の面積は　”＋ menseki ＋“です”）；

題材２　半径を入力し，円周の長さ，円の面積，球の体積を求める。
　　但し，円周率は3.14とする。

題材３　数量と単価を入力し，売上金額を求める。

題材４　a，b，c にそれぞれ数値を入力し，３つの数字の合計と平均を求める。

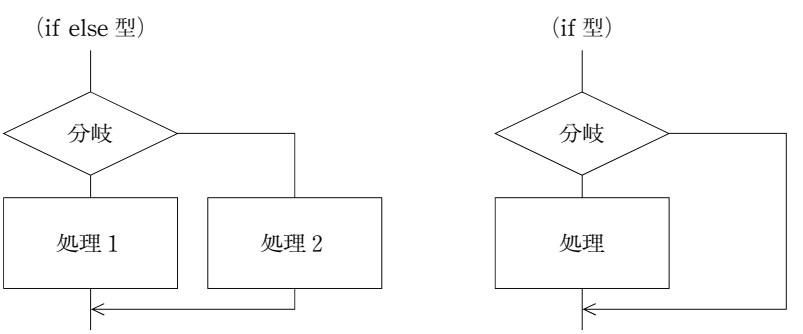
題材５　x，y，z にそれぞれ数値を入力し，x を y に，y を z に，z を x に，入れ替える（入力部分省略）。

int w＝y；　　①入れ替えロジック
y＝x；　　② w はワークエリアであり，その意味の説明
x＝z；
z＝w；

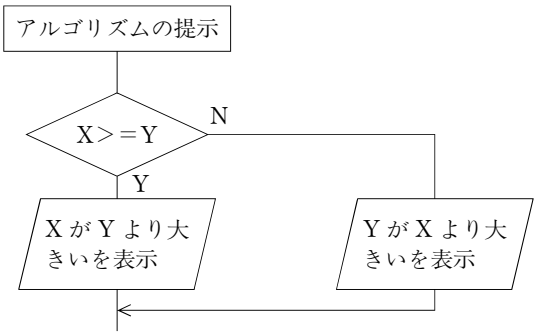
３－３　分岐型の説明と演習

１）if else 型の演習（例題の提示）

分岐型には，if else 型と if 型があることを充分説明し，プログラミングに慣れさせる。分岐のために，Java 言語では switch が用意されているが，この説明と演習を行うか否かは演習の時間数と学習者の理解度によると考えている。



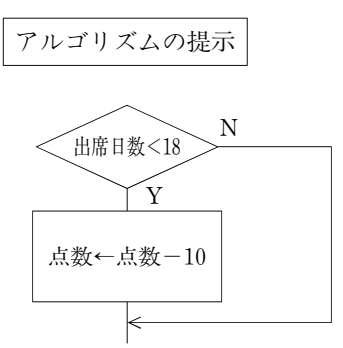
次の例題は，「２つの数字を入力し，どちらが大きいかを比較する」という例題である。これもフローチャートとともに説明し，アルゴリズムの考え方を定着させる。



Java 教材の提示	説明上強調すべき内容
BufferedReader reader＝new BufferedReader(new InputStreamReader(System.in))；	
System.out.println（“X を入れてください”）；	① X の入力を促す
String line1＝reader.readLine()；	②入力について，復習するなど折に触れて説明し定着を計る
int x＝Integer.parseInt(line1)；	
System.out.println（“Y を入れてください”）；	③ Y の入力を促す
String line2＝reader.readLine()；	
int y＝Integer.parseInt(line2)；	
if(x＞＝y){	④分岐（if else 型）について説明
System.out.println（“X＝”＋x＋“が，Y＝”＋y＋“より大きい”）；	
}	
else {	
System.out.println（“Y＝”＋y＋“が，X＝”＋x＋“より大きい”）；	
}	

2）if 型の演習（例題の提示）

if 型にも各種の題材を考察できるが，例えば「点数と出席日数をキーボードから入力させ，出席日数が18日に満たない場合は点数を減点する」などの例題の提示である。このように学習者になるべく身近な題材を用意することが重要であろう。



Java 教材の提示	説明上強調すべき内容

```
BufferedReader reader＝new BufferedReader(new InputStreamReader(System.in))；
System.out.println（“試験点数を入れてください”）；   ①試験点数の入力を促す
String line1＝reader.readLine()；
int ten＝Integer.parseInt(line1)；                    ②整数への変換
System.out.println（“出席日数を入れてください”）；   ③出席日数の入力を促す
String line2＝reader.readLine()；
int nichi＝Integer.parseInt(line2)；
  if（nichi＜＝18）{                                  ④分岐（if 型）について説明
     ten＝ten－10；
   }
System.out.println（“点数は　”＋ten＋“　です”）；
```

３）分岐型の演習（題材の提示）

題材１は分岐（if else 型）と比較演算子の扱い方，題材２（if else 型）は分岐と問題の意味の把握（ロジック）を考えさせる。題材３は分岐（if 型）と比較演算子の扱い方である。

題材１　２つの数 x，y を入力し，どちらがいくつ大きいかを表示する。
（if else 型）　但し，等しいときは x が大きいとする。

```
if（x＞＝y）{              ①分岐についての説明
    int sa＝x－y；
    System.out.println（“X＝”＋x＋“が，Y＝”＋y＋“より”＋sa＋“大きい”）；
}
else {
    int sa＝y－x；
    System.out.println（“Y＝”＋y＋“が，X＝”＋x＋“より”＋sa＋“大きい”）；
}
```

題材２　S 社のある商品の運搬費用の計算式は，次のようである。運搬距離を入力し，
（if else 型）　運搬費用を計算して表示する。

基本料金（30km まで）　　3,000円
30km を越えて１ km につき，　　80円

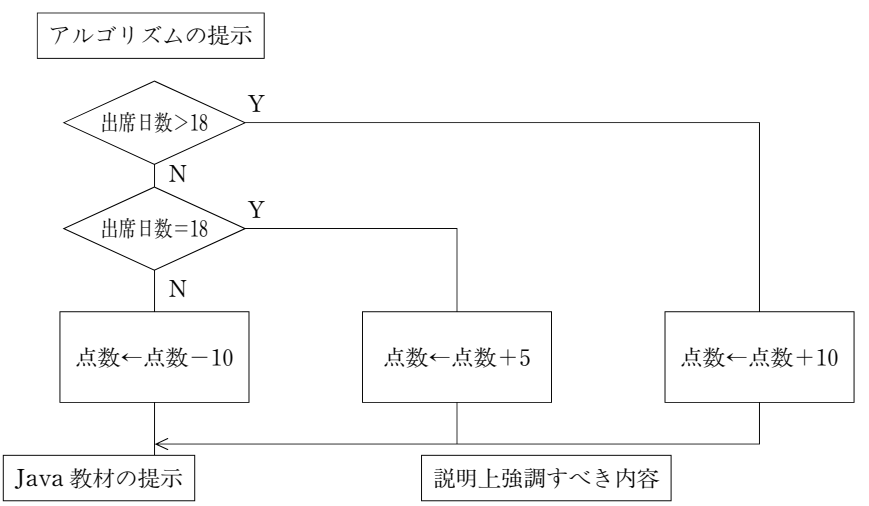
```
int hiyo＝0；
    if（kyori＞30）{                    ①分岐についての確認
        hiyo＝3000＋(kyori－30)＊80；    ②式の意味の理解
    }
    else {
        hiyo＝3000；
    }
```

System.out.println（“＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊”）；
System.out.println（“＊　運搬距離は”＋ kyori ＋“km の時 ＊”）；
System.out.println（“＊　運搬費用は”＋ hiyo ＋“　円です ＊”）；
System.out.println（“＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊”）；

題材３　２つの数字ａとｂを入力して比較し，a＞b ならばａとｂを入れ替えて，
（if 型）　a＝＜b ならばそのままで，ａとｂを表示する。入れ替えには変数ｗを使用する。

４）３分岐型の演習（例題の提示）

３分岐型にも，例えば次のような例題が提示できる。「点数と出席日数をキーボードから入力させ，出席日数が18日未満の場合は10点，18日の場合は５点減点，18日を越す場合は10点のボ




```
BufferedReader reader＝new BufferedReader(new InputStreamReader(System.in))；
System.out.println（“試験点数を入れてください”）；　　①試験点数の入力を促す
String line1＝reader.readLine()；
int ten＝Integer.parseInt(line1)；　　②整数への変換
System.out.println（“出席日数を入れてください”）；　　③出席日数の入力を促す
String line2＝reader.readLine()；
int nichi＝Integer.parseInt(line2)；
    if（nichi＞18）{　　④３分岐に関する説明
        ten＝ten＋10；　　⑤18日より多い場合
    }
    else if（nichi＝＝18）{　　⑥18日の場合
        ten＝ten＋5；
    }
    else {　　⑦18日より少ない場合
        ten＝ten－10；
    }
System.out.println（“点数は　”＋ ten ＋“　です”）；
```

ーナス点を加点する」などの題材の提示である。この段階までくると，理解が深まってプログラミングやアルゴリズム構築が面白く感じる学習者と少々重たく感じる学習者に分かれてくる。

５） 3 分岐型の演習（題材の提示）

３分岐型に関して，次のような題材を提示して定着を計る。ここまでくると，学習者はかなり慣れてくるはずである。題材１，２はワークエリア max，min，avg を求めるアルゴリズムであり，題材３はソートのアルゴリズム，題材４，５はアルゴリズムの組立と演習の題材である。

題材１　３つの数 a，b，c を入力して比較し，この内，最も大きい数を表示する。
この時，変数 max を利用する。

題材２　５つの数 a，b，c，d，e を入力して比較し，この内，最大値，最小値，平均値を求めて表示する。この時，変数 max，min，avg を利用する。

題材３　４つの数 a，b，c，d を入力し，小さい順に並べる。
この時，変数 max，min を利用する。

題材４　K 社の残業手当は次のように支給される。残業時間を入力し，残業手当を計算して表示する。但し，残業単価は，800円とする。

残業時間が	20時間以下の場合	単価の1.25倍
	20－50時間の場合	単価の1.3倍
	50時間を越える場合	単価の1.5倍

題材５　試験の点数を入力し，次のように評価して表示する。

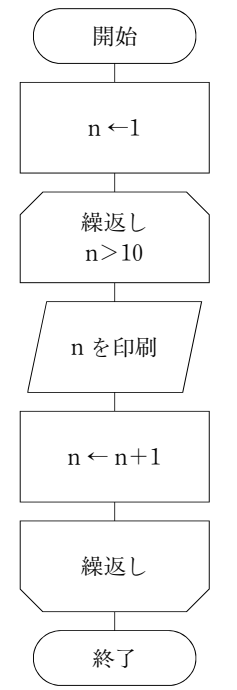
試験点数	試験評価
100－80	A
79－60	B
59－50	C
49－0	D

3－4　繰返し型の説明と演習

１）単純な繰返し型の演習（例題の提示）

繰返し型には，while 型，for 型，do-while 型がある。本稿では，while 型を中心に述べるが，他の型をどのように扱うかは，演習の時間数と学習者の理解度によろう。要は繰返しのアルゴリズムが学習者にどのように体得されるかが大切である。

単純な繰返し型には,例えば次のような題材が考えられる。「１から10までの数字を表示する」などの例題の提示である。



Java 教材の提示	説明上強調すべき内容
public class Ex300 {	
public static void main(String args[]) {	
int n＝1；	
while（n＜＝10）{	①ループの終了条件の説明
System.out.println(n)；	
n＋＋；	②ループカウントの説明
}	③ n＞10　と　n＜＝10　の意味を理解させる
}	
}	

この例のように，フローチャートにおけるループの終了は「終了条件（この例では，n＞10）」であるが，一方 Java 言語の while 文におけるループは「条件が成り立っている間（この例では，n＜＝10）」である。この意味と相違点を充分理解させる必要がある。

2）単純な繰返し型の演習（題材の提示）

題材１は繰返しアルゴリズムの演習，題材２，３はデータ入力とそれを利用した繰返しアルゴリズムの定着である。

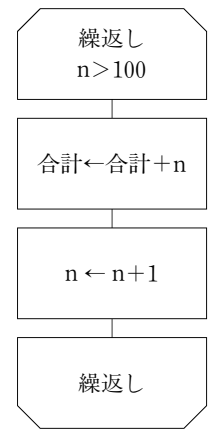
題材１　２から40までの偶数と１から39までの奇数を画面に表示する。

題材２　任意の整数 n を入力し，１から n まで表示する。

題材３　任意の整数 m と n を入力し，m から n まで表示する。但し，m＜n となるように入力する。

3）繰返し型の演習（合計値・平均値・最大値等の計算：例題の提示）

繰返しのアルゴリズムを応用し，学習者に合計値・平均値・最大値・最小値のアルゴリズムを理解させる例題である。繰返し型を利用した合計計算には，例えば次のような題材が考えられる。「１から100までの合計を求めて表示する」などの例題の提示である。



アルゴリズムの提示	Java 教材の提示	説明上強調すべき内容
	（一部省略）	
	int n＝1；	
	int gokei＝0；	
	while（n＜＝100）{	①ループの終了条件の説明
	gokei＝gokei＋n；	②合計の計算方法の説明
	n＋＋；	
	}	
	System.out.println（“１から100までの合計は” ＋ gokei ＋”）；	

４）繰返し型の演習（合計値・平均値・最大値などの計算：題材の提示）

題材１から４までは，合計を求めるアルゴリズムの定着，題材５は合計と平均を求めるアルゴリズムである。題材６は最大値，題材７は最大値と最小値，題材８は最大値，最小値，平均値を求めるアルゴリズムである。

題材１　１から20までの整数を表示し，最後に合計も表示する。

題材２　２から40までの偶数を表示し，最後に合計も表示する。

題材３　ｎを入力し，１からｎまでの合計を求めて表示する。

題材４　ｍとｎを入力し，ｍからｎまでの合計を求めて表示する。
但し，ｍ＜ｎになるように入力する。

題材５　何件かの得点を入力する。入力した件数及び得点の合計と平均を求めて表示する。但し，０が入力されたら終了する。

```
int kensu＝0；
int gokei＝0；
int heikin＝0；
BufferedReader reader＝new BufferedReader(new InputStreamReader(System.in))；
System.out.println（“得点を入れてください”）；
String line＝reader.readLine()；                    ①１件目の入力はループの
int tokuten＝Integer.parseInt(line)；                 外で行う
    while（tokuten !＝0）
    {
        kensu＋＋；                                  ②件数＝件数＋1
        gokei＝gokei＋tokuten；                      ③合計＝合計＋得点
```

```
        System.out.println（“得点を入れてください”）；
        line＝reader.readLine()；                    ④２件目以降の入力はルー
        tokuten＝Integer.parseInt(line)；              プの最後で行う
      }                                              ⑤件数＝0は，最初から０が
    if(kensu!＝0) {                                    入力された場合の対処
    heikin＝gokei/kensu；
    }                                              ⑥平均＝合計／件数
  System.out.println（“件数は”＋ kensu ＋“です”）；   ⑦件数の表示
  System.out.println（“合計は”＋ gokei ＋“です”）；   ⑧合計の表示
  System.out.println（“平均は”＋ heikin ＋“です”）；  ⑨平均の表示
```

題材６　何件かの得点を入力する。この内，得点の最大値を求めて表示する。
　　但し，０が入力されたら終了する。（以下に一例を示す：前半部省略）

```
int max＝0；                                       ①最大値＝0
int kensu＝0；                                     ②件数＝0
BufferedReader reader＝new BufferedReader(new InputStreamReader(System.in),1)；
System.out.println（“得点を入れてください”）；      ③１件目の入力
String line＝reader.readLine()；
int tokuten＝Integer.parseInt(line)；
    while（tokuten !＝0）{
    kensu＋＋；                                    ④件数＝件数＋1
    if（tokuten＞max）{
      max＝tokuten                                 ⑤最大値を保存
    }
    System.out.println（“得点を入れてください”）；  ⑥２件目以降の入力
    line＝reader.readLine()；
    tokuten＝Integer.parseInt(line)；
    }
System.out.println（“件数は”＋ kensu ＋“です”）；
System.out.println（“最大値は”＋ max ＋“です”）；
```

題材７　入力した得点の中から，得点の最大値と最小値を求めて表示する。
　　但し，０が入力されたら終了する。

題材８　全クラスの身長を入力し，クラスの人数及び身長の最大値，最小値，平均値を求めて表示する。

3－5　配列とその他の説明と演習

どのような言語を利用しようと，「配列」はプログラミング教育上必須の項目である。配列についても，Java 言語ではクラスやオブジェクトを特に意識せずに教育することができる。すなわち今まで述べてきた第１フェーズの教材の一環として提示することができる。ここでは紙面の都合上，個々の例題や題材の提示は割愛するが，次のような教材の制作を考えることができる。

１）配列の宣言　　配列の宣言方法と意味を図示する
２）配列の初期化　　配列の要素数だけの数値を代入する
３）配列の代入　　配列にキーボードから値を代入する
４）配列の利用　　作成した配列を利用する。呼び出して値を印刷する等
５）配列のソート　　配列の内容をソートする
６）多次元配列　　２次元配列等の多次元配列の教材を用意することができるが，第１フェーズで取り上げるか否かは，演習時間数による

4　Java 言語教材試案（第 2 フェーズ）

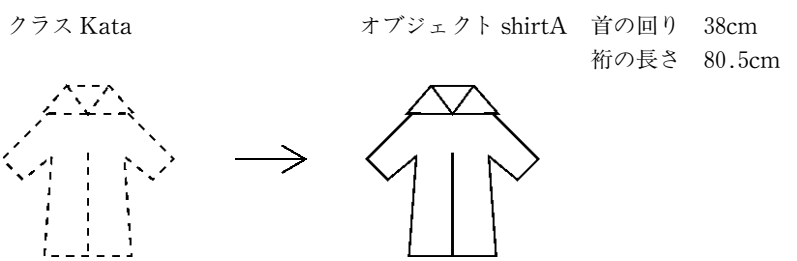
4－1　概念把握

今まで述べてきたように，Java 言語教材試案の第１フェーズは基本３構造と配列である。この段階が終了すると，学習者は Java 言語に慣れてきているし，ある程度のプログラミングやアルゴリズムを作ることに自信を持つようになろう。第２フェーズでは，オブジェクト指向プログラミング上欠かせない「クラス」と「オブジェクト」の概念把握の教材を提示する必要がある（本稿では第２フェーズは入口のみにとどめる）。

4－2　クラスとオブジェクトの関係

Java 言語プログラミング教育のためには，まず「クラス」という概念とそれから作り出される「オブジェクト（インスタンス）」を理解させるための教材の提示が求められる。学習者にとって一番難しいのがクラスやオブジェクトの概念把握であろう。この概念を把握すれば，オブジェクト指向プログラミングの半分は理解したといっても過言ではない。「クラス」を説明するには，現実にあるいろいろな題材を考えることができるが，なるべく学習者に卑近な例題をあげ，そのイメージと「クラス」「オブジェクト」の概念を重ね合わせて説明するのが得策である。現実の題材とプログラムの例題に乖離があるとかえって理解を妨げることがあるので注意が必要であろう。筆者が考えた題材は，Ｙシャツの型紙（クラス）と，これから作り出された各種サイズのＹシャツ（オブジェクト）である。型紙からシャツを作るイメージである。そのためには学習者が頭の中で描けるようなイメージ図を用意し，それに添って題材を注意深く提示･説明･演習させることが大切である。この段階でなるべく多くの演習時間を取り，「クラス」と「オブジェクト」の概念をしっかり捉えることが大切であろう。この場合，クラスは型紙，

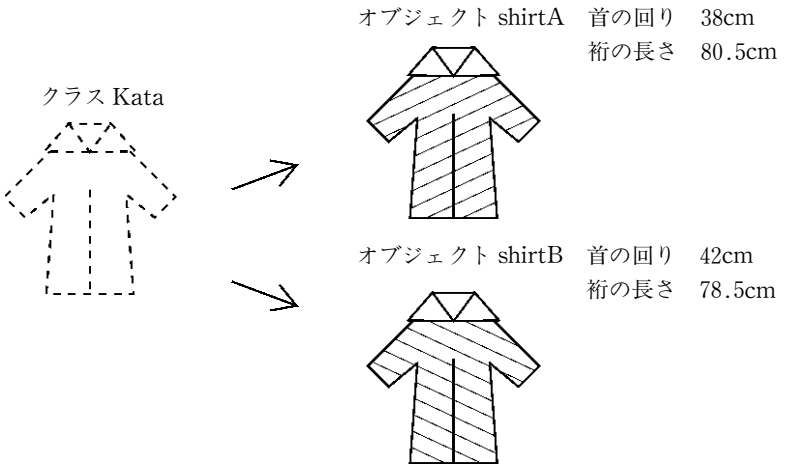
オブジェクトはＹシャツを意味するが，オブジェクト（物体の意味）とインスタンス（実例とか場合の意味）は同一の意味で使われるので，説明には言葉の統一などを心がけるべきであろう。



教材の提示	説明上強調すべき内容
//Y シャツクラス	
class Kata {	①型紙（Kata）クラスであることの強調
int kubi；	②変数の意味の復習，首(kubi)と裄(yuki)を変数とする。
double yuki；	また，これらの変数をオブジェクトと呼ぶという
}	基礎予備知識
class Sx100 {	③ここがオブジェクトを作るクラスであることの強調
public static void main（String args[]）{	
Kata shirtA；	④型紙（Kata）からシャツＡ（shirtA）を作るという宣言
shirtA＝new Kata()；	シャツＡを型紙から実際に作る宣言
	オブジェクト（物体）がここで作られる
shirtA.kubi＝38；	⑤シャツＡの首の寸法の入力
shirtA.yuki＝80.5；	シャツＡの裄の寸法の入力
System.out.println（“首の周りは”＋ shirtA.kubi ＋“です”）；	
System.out.println（“裄の長さは”＋ shirtA.yuki ＋“です”）；	
}	
}	

4－3　クラスと 2 つのオブジェクトの関係

先の例で「クラス」と「オブジェクト」の概念を理解させたが，更にその概念を定着させるためには，クラスと複数のオブジェクトを演習させることは良い結果を生む。この段階にくると，「型（クラス）」とそれから作り出された「モノ（オブジェクト）」という概念が認識されるようになってくるはずである。



教材の提示　　説明上強調すべき内容

```
//Y シャツクラス
class Kata {
    int kubi;
    double yuki;
}
class Sx101 {
    public static void main (String args[]) {
        Kata shirtA;
        shirtA=new Kata
        shirtA.kubi=38;
        shirtA.yuki=80.5;
        Kata shirtB;
        shirtB=new Kata();
        shirtB.kubi=42;
        shirtB.yuki=78.5;
        System.out.println("A さんの首の周りは" + shirtA.kubi + "です");
        System.out.println("A さんの裄の長さは" + shirtA.yuki + "です");
        System.out.println("B さんの首の周りは" + shirtB.kubi + "です");
        System.out.println("B さんの裄の長さは" + shirtB.yuki + "です");
    }
}
```

① Kata という型紙からシャツ A を作るという宣言
シャツ A を Kata という型紙から実際に作る宣言
１つ目のオブジェクト（物体）がここで作られる

② Kata という型紙からシャツ B を作るという宣言
シャツ B も Kata という型紙から実際に作る宣言
２つ目のオブジェクト（物体）がここで作られる

おわりに

本稿は基本アルゴリズムを Java 言語教育教材にどのように反映していくかを考察したものであり，本格的にクラスやオブジェクトの教育教材を取り上げたものではない。Java 言語の守備範囲の広さから考えるとほんの入口を述べただけに過ぎないが，BASIC 言語教育で培った教材とノウハウを Java 言語に展開することができたと考えている。今後，この教材を実際の授業に適用し，Java 言語プログラミング教育と教材の充実を計りたい。

インターネットに代表されるここ数年のネットワークの発達は急速であり，教材研究や教材開発が追随できないようなスピードである。従来のように，学内テキストを作成し，授業に使用して審査し，市販教材などに仕上げていくような方法では世の中のスピードに遅れをとろう。私立大学情報教育協会（私情協）も，各大学各教員が作成するある事柄に関する教材をインターネットに乗せ，相互利用する体制の整備に着手し始めた。Java 言語がネットワークを活用しつつクラスライブラリや他のクラスを「継承」して，素早くプログラムを開発していくように，教育界も個々の教員が開発した教材をネットワークを介して相互利用し，素早く教育に適用していく時代になっていくことが考えられる。

参考文献

（1） Java 言語プログラミングレッスン上　結城浩著　ソフトバンク　1999．6．
（2） Java 言語プログラミングレッスン下　結城浩著　ソフトバンク　1999．7．
（3） Java①　桑原信也著　翔泳社　2001．1．
（4） Java②　桑原信也著　翔泳社　2001．2．
（5） Java 入門教室　丸の内とら著　翔泳社　2001．2．
（6） Java プログラミングブック　河西朝雄著　技術評論社　2001．6．
（7） 初体験 Java　丸の内とら著　技術評論社　2000．8．
（8） 第 3 版 Java プログラミング　中島省吾著　SCC　2001．3．
（9） Java によるプログラミング入門　久野禎子他著　共立出版　2001．6．
(10) チャレンジ Java　新居雅行著　ローカス　2000．6．
(11) Java 入門早わかり　堀桂太郎著　オーム社　2001．4．
(12) Java プログラミング　立木秀樹他著　共立出版　2000．10．
(13) Java のすべて　山下関哉著　日本実業出版　2000．9．
(14) Java がわかる　藤田一郎著　技術評論社　1999．12．
(15) 携帯電話 Java　橋本賢一著　リックテレコム　2001．7．
(16) Java による図形処理入門　山本芳人著　工学図書　1998．3．
(17) はじめての i モード Java プログラミング　コネクト著　日経 BP　2001．4．
(18) Java の達人　田中成典編集　森北出版　1999．12．
(19) Java 教科書　岡田謙一他著　ソフトリサーチセンター　1999．11．
(20) Java 集中ゼミ　泉田洋一著　ソフトバンク　2001．6．
(21) ザ・Java　戸川隼人著　サイエンス社　1998．4．
(22) スタートアップ Java　技術評論社　2001．6．
(23) Java によるはじめてのアルゴリズム入門　河西朝雄著　技術評論社　2001．7．
(24) やさしい Java　高橋麻奈著　ソフトバンク　2000．9．
(25) Java ゲームアプレット　村山要司著　工学社　2001．7．
(26) Java で作る i アプリ　河西朝雄著　ナツメ社　2001．7．
(27) BASIC で学ぶ　フローチャート技法　海老澤信一他著　啓学出版　1989．

（注）本稿の執筆分担は次のようである。

海老澤　信一　　はじめに，おわりに
　　3　Java 言語教材試案（第1フェーズ）
　　4　Java 言語教材試案（第2フェーズ）

矢野口　聡　　1　Java 言語の生い立ち
　　2　Java 言語プログラミング教育の導入