

Title	文系学生へのプログラミング教育
Author(s)	国分, 道雄
Citation	聖学院大学論叢,20(2) : 197-206
URL	http://serve.seigakuin-univ.ac.jp/reps/modules/xoonips/detail.php?item_id=42
Rights	

聖学院学術情報発信システム : SERVE

SEigakuin Repository for academic archiVE

文系学生へのプログラミング教育

国分道雄

Programming Education for Non-Science Majored Students

Michio KOKUBUN

Key words: Programming Education, Non-Science Majored Student, Java

1. はじめに

学生に対して、どのような「情報教育」を行うべきであろうか。

特に「コンピュータプログラミング教育」に関しては、時代の流れと共に位置付けが大きく変化してきている。「情報処理教育」と呼ばれていた頃には教える内容として「プログラミング」以外の選択肢は無かったが、現在のように「情報教育」に様々な題材がある状況で「プログラミング」を扱う意義を見直す動きがある。高等学校の新教科「情報」では、「プログラミング」の扱いは大きくないが、基本的なアルゴリズムについては取り扱う科目もある。

このような中で、本学のように文系の学生に対する「プログラミング教育」の指導を行う場合について考察したい。

2. 情報処理教育から情報教育へ

現在では、「情報教育」という言い方が一般的になっているが、1980年代までは主に「情報処理教育」という言い方をされていた。⁽¹⁾ 本来、「情報処理」(Information Processing)とはコンピュータを利用しない場合にも使う言葉であるが、主に「コンピュータプログラムによるデータ処理」のことを指す用語として使用された。当時は、コンピュータを扱うにはプログラミングが必須だったこともあり、「情報処理教育」とは「プログラミング教育」のことであった。この時代に、文系の学生にも「プログラミング教育」を行っていたのは、「コンピュータを教えるにはプログラミングを教えるしかないから、文系学生に対してもプログラミング教育を行う」という理由からである。

1980年代後半になると、日本語ワープロソフトや表計算ソフト等のアプリケーションソフトが普及し始め、プログラミング言語を習得しなくてもコンピュータを利用することができるようになってきた。1988年には東京大学教養学部で実験的に、プログラミング教育をせずに、コンピュータの操作とデータ解析の手法を学ばせる授業も行われている。⁽²⁾ ここでは、従来の「情報処理教育」の大きな柱であるプログラミング教育を否定はしないが、プログラミングでよく使われる練習問題が文科系の教育内容と乖離しているため、「コンピュータ嫌い」の学生を増やす原因になっていることを指摘している。文科系独自の情報処理教育を構成するために、その目的を「道具としてのコンピュータ」「手法としての情報システム」「目的としての情報化社会」の3つに集約し、

(1) 情報処理の基礎

コンピュータを使用しないものも含む

(2) 基本操作の実習

キーボードやアプリケーションソフトの操作等

(3) 専門教育としての情報処理論

各専門分野の実際のデータ処理

(4) 社会情報システム

現代社会の各種情報システムの意義と問題点

という4段階のカリキュラムが提案されている。

このような動きの中で、1990年前後には、従来のプログラミング教育を中心とした「情報処理教育」とは違う形の教育が模索され、⁽³⁾ 従来の「情報処理教育」に対して「一般情報教育」や「情報基礎教育」などと呼ばれるようになる。これが、広く「情報教育」と呼ばれるようになる過渡期でもあった。教育内容にプログラミング以外の選択肢が出来てきたことを受けて、文系の学生に対しては負担になるプログラミング教育を避けるところが増えてきた。

一方で、原田(1992)⁽⁴⁾のように、文系学生への情報教育としてプログラミング教育を行う意義を見直し、従来の理由とは全く異なる考え方から、文系学生にも積極的にプログラミング教育を行う動きも起きた。そこには、アプリケーションシステムを道具として使用する場合でも、言われた操作を繰り返すだけで「自分が今何をしているのか」も分からない「不健全なユーザ」ではなく、システムを自分の問題解決のために主体的に利用とする態度とシステムを道具として使いこなす能力を持つ「健全なユーザ」を育成するという考え方がある。システムを道具として使いこなすには、システム内部の概念モデルを把握することと、自分の操作に対して統制感を持つことが必要であり、特定のアプリケーション利用経験だけでは表面的な変化にとらわれてしまうため概念モデルの獲得が難しく、プログラミングを通じた学習の方が有益であるとしている。ただし、「プログラミングを強制することによりコンピュータアレルギーを起こす」との反省に立ち、理工系の学生へのプログラミング言語教育とは異なり、特定の言語の仕様をマスターしてプログラムが書けるようになる

ことを目的とはしない。また、従来の数学的な題材とは異なる教材を検討していく必要性が言われている。

3. 高等学校「情報」とプログラミング

2003年度より高等学校では、新しい教科である「情報科」が始まった。新教科「情報」は、高校普通科で必修となった普通教育に関する教科「情報」と、「商業」「工業」等と同等の専門教育に関する教科「情報」の2つがある。⁽⁵⁾

学習指導要領⁽⁶⁾で「プログラミング」について触れているのは、普通教科「情報」では3科目のうち「情報B」1科目、専門教科「情報」では11科目のうち「アルゴリズム」「情報システムの開発」の2科目である。

普通教科「情報」の科目「情報B」は、「情報教育の3つの観点」のうち「情報の科学的な理解」を重視した科目である。内容は、大きく4つに分かれていて、

- (1) 問題解決とコンピュータの活用
 - ア 問題解決における手順とコンピュータの活用
 - イ コンピュータによる情報処理の特徴
- (2) コンピュータの仕組みと働き
 - ア コンピュータにおける情報の表し方
 - イ コンピュータにおける情報の処理
 - ウ 情報の表し方と処理手順の工夫の必要性
- (3) 問題のモデル化とコンピュータを活用した解決
 - ア モデル化とシミュレーション
 - イ 情報の蓄積・管理とデータベースの活用
- (4) 情報社会を支える情報技術
 - ア 情報通信と計測・制御の技術
 - イ 情報技術における人間への配慮
 - ウ 情報技術の進展が社会に及ぼす影響

となっている。「(4) 情報社会を支える情報技術」で「情報社会に参画する態度」も育成する。また全体を通して、「情報活用の実践力」も身に付けさせる。「(2) コンピュータの仕組みと働き」の「イ コンピュータにおける情報の処理」の中で、「コンピュータの仕組み、コンピュータ内部での基本的な処理の仕組み及び簡単なアルゴリズムを理解させる。」としている。ただし、内容の取扱いについては「コンピュータ内部での基本的な処理の仕組みについては、一つ一つの命令がステップで動いていることを扱う程度とする。アルゴリズムの具体例については、並べ替えや探索などのうち、

基本的なものにとどめるようにする。」としていて、プログラミング言語によらない指導も可能である。「(3) 問題のモデル化とコンピュータを活用した解決」の内容の取扱いについて、「ソフトウェアやプログラミング言語を用い、実習を中心に扱うようにする。その際、ソフトウェアの利用技術やプログラミング言語の習得が目的とならないようにする。」としていて、プログラミング言語の実習を行うが、その習得が目的とならないように強調されている。あくまで、問題をモデル化し解決する手段として、プログラミング言語を利用するという考え方である。このように、普通教科「情報」の指導では、プログラミングに深入りしない方針になっている。

専門教科「情報」の科目「アルゴリズム」の内容は、

- (1) 数値計算の基礎
 - ア 基本的なアルゴリズム
 - イ 数値計算
- (2) データの型とデータの構造
 - ア データの基本的な型と構造
 - イ データ構造とアルゴリズム
- (3) 整列
- (4) 探索
- (5) データベースの概要
 - ア ファイルとデータベース
 - イ データベースの仕組み
 - ウ データベースの設計と操作

となっている。内容の範囲や程度については、「基本的なアルゴリズム」について「アルゴリズムとプログラムに関する基本的な内容を扱い、順次、選択、繰り返し構造で表現できるアルゴリズムについて理解させること。」としている。「データの基本的な型と構造」について「数値型、文字型及び論理型並びにレコード及び配列を扱うこと。」としていて、配列も扱うことが明記されている。「整列」については「複数の基礎的な整列法」、「探索」については「線形探索法と二分探索法」を取り上げ、「それぞれの基本的な考え方、具体的なアルゴリズム及びその違いについて理解させ、効率的なアルゴリズムについて考えさせること。」としている。専門教科だけあって、普通教科とは大きく異なり、プログラミングの具体的な内容に踏み込んだ指導を行うようになっている。特に、「情報B」の内容の取扱いでも触れられていた「並べ替え（整列）」と「探索」については、複数の方法を比較させるようにしている。

専門教科「情報」の科目「情報システムの開発」の内容は、

- (1) 情報システムの概要
 - ア 情報システム化の技法

イ ソフトウェア開発の基礎

(2) 情報システムの設計

ア プログラム設計

イ プログラミングと単体テスト

(3) ソフトウェアテスト

(4) 運用保守

となっている。科目の指導にあたって、「開発する情報システムに応じて適切なプログラム言語を選択し活用できる能力の育成に留意すること。」としていて、選択できるように複数のプログラミング言語を取り上げることが示唆されている。「情報システムの設計」については、「構造化設計の考え方について理解させること。なお、オブジェクト指向設計も、生徒の興味・関心に応じて扱うことができること。」としていて、生徒に関心があればオブジェクト指向言語を使った設計も扱う場合がある。

情報科教員免許は、普通教科「情報」と専門教科「情報」の両方を含んでいる。したがって、情報教職課程を履修する学生は、文系学生であっても、これらの内容を指導できる力を身につける必要がある。そのためには、「並べ替え」「探索」等を含むプログラミング教育を受けることも必要になる。

4. 教育に用いるプログラミング言語

「情報処理教育」が始まった頃には、FORTRAN, COBOL 等のプログラミング言語を用いたプログラミング教育が行われていた。これは、科学技術計算やビジネスの現場で実際に使用されていた実用言語であった。

1980年代には、パソコンの普及により、初心者向けの BASIC が広く使われるようになった。小学校等では、亀に命令を与えて絵を描かせる「タートルグラフィックス」によってプログラムの概念を理解しやすい LOGO を用いて指導しているところもあった。いずれも、実用性よりも、初心者にも学びやすいことを目的とした教育用言語であった。専門的な教育では実用言語による指導、入門的な教育では教育用言語による指導が行われる例が多かった。

その後、システム開発などの現場では実用言語として C 言語等が使われることが多くなり、教育用言語としても Pascal のような構造化言語が用いられることもあった。一方で、BASIC の機能も増えて、さらに BASIC の教育を受けた学生が社会に出て行くようになった影響もあり、実際の現場で使われるようにもなった。

現在の実用言語としては、オブジェクト指向も取り入れた C++, Java, JavaScript, Visual Basic 等が挙げられる。

最近の教育用言語としては、Squeak を用いて小学校等で指導しているところもある。⁽⁷⁾ 日本語でプログラミングできる「なでしこ」という言語もあり、初心者向けでありながらファイル操作や Microsoft Word/Excel との連携も行うことができる。⁽⁸⁾ 大学入試センター試験の「情報関係基礎」で出題されている日本語による手順記述言語に準拠した PEN という初学者向けプログラミング学習環境もある。⁽⁹⁾ 簡単なスクリプトを記述するだけで手軽にソフトを作成できる HSP を用いて、グラフィックやサウンドを使ったゲームを作ることもできる。⁽¹⁰⁾

兼宗 (2007)⁽¹¹⁾ は、教育用のプログラミング言語に求められる性質として「プログラム記述の工夫 (構文の簡略化等)」「適切なフィードバック (実行結果の視覚化等)」「興味と達成感を持つ題材 (ゲーム作成が可能等)」を挙げている。

5. 文系学生への「プログラミング教育」の留意点

2003年度より、本学コミュニティ政策学科の専門科目「情報システム論B」を担当させていただいており、この中で「プログラミング」も取り扱っている。

受講学生には、情報科教員やIT系企業への就職を目指している学生がいる一方で、タイピングや表計算ソフトの操作も苦手な学生もおり、スキルの差は大きい。

そのため、特定の言語の細かい仕様に深入りせず、必要最小限の文法に留める方針をとった。理工系の学生であれば「与えられた課題を解決するプログラムを独力で書けるレベル」が要求されるが、本授業では「自力で書くことは出来なくても、読むことは出来るレベル」を目指した。スキルの高い学生に対しては、自力でプログラムを書けるように指導していくが、その場合でも、まずはプログラムを読む力が必要になる。

単純なものであっても、プログラミングを実際に体験すれば、例えば「たった1文字の入力ミスでも、意図しない動作になってしまう」等のプログラムの特性を経験し、よく遊んでいるゲームソフトやアプリケーションソフトのように巨大なプログラムを作成する苦労や現代社会で利用されている情報システムにもエラーがある可能性にも気づくことができる。この経験を通して、ソフトの違法コピーを行わない等の著作権を尊重する態度や情報社会で生きる上で留意すべき事柄を体得することを目的としている。

言語の仕様を習得させることが目的ではないので、プログラムの表記にも配慮が必要である。例えば、よく使われる「1からnまでの和を求めるプログラム」は、CやJavaなどでは、

```
int n= 10, sum= 0;
for ( int i= 1 ; i<=n ; i++)
    sum+=i;
```

と書ける。しかし、本授業では少しでも学生が理解しやすくなるように配慮して、これを例題とし

て用いるときには、

```
int goukei= 0;
for ( int k= 1 ; k<= 10 ; k=k+ 1 ) {
    goukei=goukei+k;
}
```

のように書いている。プログラムに慣れていない学生にとっては使用する変数が多いと難易度が上がってしまうので、ここでは変数 n を使わない書き方にした。「 $i++$ 」「 $sum+=i$ 」のような表記も避け、for で繰り返す命令文が 1 行でも「 $\{ \}$ 」で囲むようにしている。特殊なケースの省略方法を教えるより、多くの場合に使える方法で記述し、文法の説明を少なくする配慮である。また、 i と 1 は見間違いやすいため、「変数 i 」の使用も避けている。

6. コミュニティ政策学科の授業での実践例

最初の2003年度には、VBA (Visual Basic for Applications) , Visual Basic, C++ を扱った。導入として VBA でプログラミングの概念に触れて、慣れてきたところで Visual Basic で次の段階への橋渡しをして、最終的に実用言語である C++ で Windows プログラミングを体験することを計画した。このように段階をおいたのは、C++ による Windows プログラミングでは、多くの宣言命令を記述することが必要になるため、プログラム初心者がいきなり触れると本質の部分が見えなくなってしまうからである。VBA は Microsoft Excel を利用し、C++ の開発環境はオープンソースの WideStudio⁽¹²⁾ を利用したため、学生に自宅で使用させることも可能であった。しかし、Visual Basic の開発環境には Microsoft Visual Studio を利用したため、課題の続きを実習室外で行うことが難しかった。また、3つの環境を利用することも、その度に操作方法を覚えねばならないため、学生の負担が大きいように感じた。一方で、Windows プログラミングの「画面をクリックすると変化する」「ウインドウ内を画像が動く」という課題には学生の関心が高く、「数学的な題材で、計算結果が数字で表示される」という課題よりも興味を持って積極的に取り組んでいた。

次の2004年度は、前年度の反省から Visual Basic はやめて、VBA と Processing⁽¹³⁾ の2つを扱った。導入の VBA では、条件分岐構造 (if) と繰り返し構造 (for) を扱う程度に留めて、これらの基本概念を理解させることを目的にした。Processing は Java ベースの言語で、基本的な文法は Java と共通である。グラフィック関係の命令が手軽に使えるようになっていて、宣言命令無しにウインドウ上に図形を描き動かすことが出来るため、デザインやアート面で多く利用されている。Processing は開発環境でもあるので、プログラムの入力からデバッグ・実行まで出来て、さらに Java としてのソースプログラムやアプレット (class ファイル) を出力することも出来る。最終的には、Processing で Java アプレットのゲームを作成させた。ゲームとして遊べるプログラムを目

標にしたため、ハードルは高くなったが、学生は興味を持って取り組んでくれた。しかし、導入の VBA (BASIC) と Processing (Java) では、文法が大きく異なるため、戸惑う学生が見られた。また、Processing はベータ版であったため、動作が不安定になることがあったり、日本語が使えなかったりする面は扱いにくかった。

2005年度は、前半は JavaScript を扱い、後半は Java を扱った。JavaScript と Java は全く異なる言語であるが、基本的な文法は共通である。そこで、基本概念 (条件判断, 繰り返し, 関数, 配列等) の学習は、JavaScript で行うことにした。プログラムの入力 は Windows 標準のメモ帳で行い、実行は Web ブラウザ (Internet Explorer) で行ったため、開発環境は利用しなかった。後半の Java では、アプレットでゲームを作成した。Java の開発環境には、オープンソースの Eclipse を利用した。Eclipse 上のエディタには、「文法エラーの箇所に赤い下線が表示される」等の機能があり、慣れていない学生が自分の入力ミスを見つけるのが容易であった。それに対して、JavaScript をメモ帳で入力していた時には、当然そのような機能はなく、1文字の入力ミスがあるだけで、実行した Web ブラウザには何も表示されないことも多く、ミスを探すのが困難であった。

2006年度は、Java のみを扱った。ただし、ゲームとして動作するような Java アプレットのソースプログラムは、宣言命令などだけで数十行になってしまうため、前半は Java アプリケーションのプログラムを扱うことにした。Java アプリケーションであれば、

```
public class p1 {
    public static void main (String[] args) {
```

ここにプログラムを書く。

```
}
}
```

というように、最初に2行の宣言命令と最後に2つの括弧が必要になるだけなので、導入として利用することも問題がなかった。また、最初から Eclipse を使用したので、入力ミスを見つけやすくなった。後半は、前年度と同様に Java アプレットでゲームを作成した。文法的な断絶が無いため、学生の負担は減ったように感じられる。ただ、Java アプリケーションと Java アプレットでは、Eclipse 上での実行時の操作に少し異なる部分がある。(Java アプリケーションの実行は「実行ダイアログ」で「Java アプリケーション」を選ぶが、Java アプレットの実行は「実行ダイアログ」で「Java アプレット」を選ぶ。)一部の学生ではあるが、Java アプリケーションから Java アプレットの切り替え時に実行操作で戸惑う者がいた。

現在の2007年度は、最初から Java アプレットのみを扱っている。Java アプリケーション的に扱うのであれば、アプレットであっても、

```
import java.applet.Applet;  
public class p1 extends Applet {  
    public void init () {
```

ここにプログラムを書く。

```
    }  
}
```

のように短く書くことができる。Java アプリケーションの場合に比べて、1行目の「import 命令」と「extends Applet」の部分が增えているが、「public static void main (String[] args)」が「public void init ()」と短くなっているのだから、導入にはこちらでも大丈夫であった。これにより、導入からゲーム作成まで Java アプレットで完全に連続するようになった。その詳細については、別途報告している。⁽¹⁴⁾

7. おわりに

プログラミング教育をめぐる状況について、概観した。

文系学生へのプログラミング教育は、理工系学生に対するものとは違った教授法が必要となる。しかし、段階ごとに十分な配慮をし、興味を惹く題材等について工夫をすれば、簡単なゲームプログラムを理解できるレベルまで伸びていく。

また、情報社会で生きる力を育む上でも、プログラミングの経験は極めて重要である。今後とも、学生達が社会で生きていく力を身に付けられる指導を検討していきたい。

注

- (1) 御牧義「大学等における一般的情報処理教育の実態について」『情報処理学会研究報告「コンピュータと教育」』Vol. 1988 No. 75 1988-CE-002 1988年10月 pp. 1-6.
- (2) 常盤洋一「大学文科系における情報処理教育」『情報処理学会研究報告「コンピュータと教育」』Vol. 1988 No. 93 1988-CE-003 1988年12月 pp. 1-6.
- (3) 例えば、泉本利章「文系学生向けコンピュータ入門教育と実習環境 — 立教大学一般教育部の場合 —」『情報処理学会研究報告「コンピュータと教育」』Vol. 1991 No. 68 1991-CE-017 1991年7月 pp. 1-8.
- (4) 原田悦子「文系学部におけるプログラミング教育の意義 — 健全なユーザ育成のための情報教育の視点から —」『社会労働研究』第38巻 第3・4号 通巻第129・130号 1992年3月 pp. 119-134.
- (5) 国分道雄「高等学校における新教科「情報」の現状」『聖学院論叢』第18巻 第2号 2006年3月 pp. 61-72.
- (6) 文部科学省『高等学校学習指導要領解説 情報編』開隆堂出版 2000.
- (7) 『杉並区和田小学校でのスクイークを使った活動』<http://squeakland.jp/wadasyou/wadasyou.html>
- (8) 『日本語プログラム言語「なでしこ」公式ページ』<http://nadesi.com/>

- (9) 『初学者向けプログラミング学習環境 PEN (Programming Environment for Novices)』
<http://www.media.osaka-cu.ac.jp/PEN/>
- (10) 『HSP3 HOT SOUP PROCESSOR』 <http://hsp.tv/>
- (11) 兼宗進「教育用プログラミング言語の動向」『情報処理』 Vol. 48 No. 6 2007年6月 pp. 589- 593.
- (12) 『WideStudio/MWT Home Page』 <http://www.widestudio.org/>
- (13) 増井俊之「インターフェイスの街角 (78) — Processing —」『UNIX MAGAZINE』 第19巻 第8号
通巻214号 2004年8月 pp. 171- 176.
- (14) 国分道雄「文系学生への Java プログラミング教育」『第2回パーソナルコンピュータ利用技術学会
全国大会』 2007年12月2日にて発表。