

テーマ：「プログラミング教育を見直す」  
高等学校と大学との連携を視野に置いて

日時：2003年10月18(土) 13時30分～17時

会場：大学生協会館（東京・杉並）

報告：

1. 「高校普通科の情報教育と大学の一般情報教育におけるプログラミング教育の在り方について考える」  
綾 皓二郎（石巻専修大学）
2. 「道具性を重視したプログラミング教育」  
山田 祐仁（京都工芸繊維大学）

質疑応答：

司会：武沢 護（神奈川県立厚木南高等学校）  
（敬称略）

#### 講演要旨（1）

「高校普通科の情報教育と大学の一般情報教育におけるプログラミング教育の在り方について考える」

#### はじめに

2003年度から、高校で普通教科『情報』の授業がはじまり、2006年度から大学では現在行われているような「情報リテラシー教育」は不要になると言われており、大学には大学教育にふさわしい情報教育が行えるよう今から準備が求められている。いま授業のはじまった高校の教科『情報』と準備している大学の一般情報教育が共有すべき「情報教育」としての「プログラミング教育」についての考えを述べる。

#### 高校の情報教育におけるプログラミング教育と問題解決

高校の『数学B』では、簡単な数値計算アルゴリズムを理解し、それを科学技術計算用のプログラミング言語などを利用して表現し、具体的な事象の考察に活用できるようにしているが、選択科目であるため、旧課程と同様にきわめて限られた生徒しか、プログラミングを学ばないであろう。また、『情報B』では、問題のモデル化とコンピュータを活用した解決でソフトウェアやプログラミングを用いるとしているが、プログラミング言語そのものの習得は目的とせず、擬似言語を用いたアルゴリズムの学習に留まっている。

その一方で、コンピュータの利用に関しては、「問題解決(problem solution)」という言葉が、安易に用いられすぎており、特に『情報B』では頻出している。このように高校の情報教育（あるいは大学の一般情報教育）で「問題解決のためのプログラミング教育」を唱えるのは、やや誇大ではないか。「問題解決支援のためのプログラミング教育」あるいは、「問題解決のセンスを養うためのプログラミング教育」とすべきであろう。なぜなら、コンピュータの大部分の役割は、人間が問題を解決するための支援であるからである。つまり、コンピュータを用いて解決できる問題は、数値計算や文字・画像の処理、検索・並び替え、データ転送、さらに、その組み合わせなど、きわめて限られている。現実に直面する課題の大部分は、白か黒かで割り切れない「アナログ」の問題であり、デジタルコンピュータを使用すれば、これらを容易に解決できるような印象を、生徒に与えるべきではない。コンピュータの役割を、過大評価して教えることは好ましくない。

#### クッキングとのアナロジーから見たプログラミング

ここで、視点を変えて、調理（クッキング）とプログラミングを比較してみると、両者には次のような共通点が見いだせる。ともに、材料（入力データ）を揃え、道具（コンピュータ）を使い、手順（アルゴリズム/プログラム）に

従って、時間をかけて目的のもの（出力データ）を作る。また、そのときの過程や結果において、実益・楽しさ・科学を備えているし、関わる人も、初心者から鉄人まで、アマからプロまでと、非常に広くレベルも様々である。そこで、さらに、クッキングとのアナロジーからプログラミングを考えてみる。

まず、クッキングは、レシピ（手順）を書くことでは完結しない。実際に調理して、自分で評価しないと、目的が達成されたかどうか分からない。クッキングの楽しみも生まれてこない。つまり、結果を自ら評価することで、理解を確実にし、調理能力を身に付けることができる。もし、クッキングのためのレシピを作ることに留まっていたとするなら、生徒は満足するだろうか。

クッキングとレシピの関係と同じく、プログラミングも、アルゴリズムを擬似言語で書くことでは完結しない。プログラミングは、プログラムをプログラミング言語で書き、実際に実行させないことには、評価ができず、プログラミングは完結しない。そうすると、『情報B』でのプログラミングの場合には、生徒はプログラムを作成して実行させなくてもよいのか。アルゴリズムを書くだけで、生徒は楽しいと思うだろうか。また、生徒の理解は深まるか、運用能力は身に付くだろうか。このようなアルゴリズムに留まる学習とは、実は生徒から達成感や楽しみを奪った学習ではないのだろうか。

将来、クッキングはどうなるかと言えば、総菜や料理の外注化・商品化が、かつてないほど進んでいるため、衣や住のように普通の個人や家庭では、自ら料理をするのではなく、生存を維持するという目的は、既製品で達成し、快適性（食の場合はおいしさ）や自己表現性・楽しさを、求める方向に進むのではなからうか。そうならば、学校教育では、プロの料理人や栄養士の養成だけとなり、普通の生徒のクッキング実習は、調理済み食品を組み合わせる簡単な調理法の習得だけとなってしまわないか。

このことを、プログラミングとアルゴリズムにあてはめて考えるならば、コンピュータの実習は、「アプリケーションソフトの利用法の習得でよいのか」が問題とされるべきである。現在、プログラミングの外注化や商品化が普通になっており、また、市販のアプリケーションソフトを使うことで、たいいていのことはできる。したがって、仕事の目的は、自ら制作したプログラムでなく、既製品を利用することで達成できるので、プログラミングは、快適性や自己表現性、楽しさを求める方向に進んでいるのではないか。言い換えれば、将来、普通の個人や研究室では、自らプログラミングをすることはほとんどなく、商品化が難しいもののプログラミング、例えば、少数のもの、特別なもの、趣味的なもの、あるいは特別な研究用、障害者用などが残るのではないだろうか。そうすると、学校教育での調理実習と同じように、プログラミング教育は、プロのプログラマーやシステムエンジニアの養成だけとなってしまわないか。また、それで、良いのかが問題となる。

### 大学の一般情報教育におけるプログラミング教育

プログラミング教育の目的は、コンピュータの基本的な働きおよび「コンピュータ科学」をよりよく理解させるとともに、コンピュータに対して主体的・能動的に接する経験をもたせることや、ソフトウェアの制作から使用・保守までの全過程を体験させて、ソフトウェアの特性をよりよく理解させることである。この目的を達成するためのプログラミング教育では、プログラミングとコンピュータの動作原理との関連や、プログラミングの文法や技法の詳細よりも、構造化・抽象化・関数などのプログラミングの基礎概念をきちんと教えることが大事である。プログラミングを通じて「コンピュータ科学」の基礎概念を教えることである。

ここで、ACMの言う頻出概念から、「コンピュータ科学」の基礎概念を構成すると、再利用、バインディング、抽象化のレベル、概念的・形式的モデル、トレードオフとその結果、効率、進化、時間的順序、空間的順序、大規模問題の複雑さ、整合性と完備性、セキュリティなどがあげられるが、ブラックボックス化、段階的詳細化なども重要な基礎概念と言える。

### 入門教育用プログラミング言語の評価基準

プログラミング言語Cを学ぶと、コンピュータの動作原理やOSなどのシステム記述がよくわかることや、構造化プログラミングができ、表現力に優れ、多様なアルゴリズムを自然に、しかもコンパクトに表現できるなど多くの有利な点を持っている。

しかし、ここでは、次に示す評価基準から、入門教育用プログラミング言語を選ぶ。

- (0) 構造化プログラミングができる言語
- (1) 高校と大学のプログラミング教育の円滑な接続への配慮
- (2) 敷居が低く、教えやすく、学びやすい言語
- (3) プログラミング学習に発展性がある言語。
- (4) コンパイラのコストパフォーマンス

高校では表計算ソフトの使い方は履修済みと考えられるので、評価基準(1)に照らして、Excelとの連携が優れているExcel VBAを用いることは、合理的である。

表計算ソフトとVBAを用いたプログラムは、互いを補うものとして使える。Excel VBAを評価基準(2)にあてはめると、Excel VBAは、ワークシートを中心としてプログラミングするので、メモリを頭に描いてプログラミングするよりは、簡単に処理結果を目で確認でき、結果のグラフ化もExcelの機能を使えば容易である。また、Excel VBAは、構造化Basicであるので、学びやすい。

評価基準(3)について言えば、Excel VBAでプログラミングを学ぶことは、マクロの利用を容易にし、また、他の高級言語を学ぶ準備ともなる。

評価基準(4)のコストパフォーマンスについては、Excelを備えれば、あらためてコンパイラを購入する必要がないので、余計な費用や手間がかからない。学生は教室と自宅の両方で学習できる点でも、コストパフォーマンスに優れていると言える。

### 擬似言語の導入と使用の問題点

情報処理学会編『大学等の一般情報処理教育調査報告書』によれば、プログラミング教育の導入部で、「擬似言語」を用いて教えることが推奨されている。しかし、これには、幾つかの問題点を指摘することができる。まず、自然言語と擬似言語・プログラミング言語との違いを説明しないまま、擬似言語を導入している点や、擬似言語の仕様があいまいであること、特に入門用に最低限必要なことに絞るべきところが、絞りきれていないことである。次に言えることは、条件分岐や繰り返しで、多重の制御構造がなぜプログラミング言語には必要となるかの説明がなく、構造化についての言及もほとんどない。さらに、手続き(関数とサブルーチン)は、抽象化・構造化・再利用・効率などのコンピュータ科学の重要な概念を理解する上で必須の概念であり技法であるが、その手続きの概念と技法を教えることについても、ほとんど触れられていないなど、擬似言語の使用上の問題点は、少なくない。

繰り返しになるが、プログラミングは、アルゴリズムをプログラミング言語で書いて、実際に実行させて初めて評価ができ完結するものである。したがって、アルゴリズムを擬似言語で書くことでは、プログラミングは完結しない。つまり、擬似言語によるプログラムの作成で留まり、実行させることができないプログラミング教育では、その目的を達成することができない。

大学での一般教育という観点から情報教育に使える総時間数を考えると、教えるべき内容を取捨選択し精選しなければならない。現実問題としても、擬似言語プログラミングを一般情報教育で最初に学ぶことは、高校と大学で二度手間であり冗長である。さらに、アルゴリズムの理解においても、学習者の頭の中だけで実行をトレースする机上デバッグは、人間を機械に合わせる作業であり、あまり効果的であるともいえないし、効率的でもない。他方、学生の挫折感を回避するために、擬似言語プログラミングを教えるという考えは、擬似言語を採用する根拠としては、十分でない。むしろ、コンピュータの動作を理解させるためのプログラミング教育を目指すとき、擬似言語を用いることが、逆効果になることがある。プログラミング言語の文法の煩雑さを回避するために擬似言語を使うというのも、擬似言語を採用する理由としては弱い。

アルゴリズムの学習では、個々の定石的な手順やパズル的な問題の解法を学ばせることよりも、次に示すアルゴリズムの基本的な考え方を教えるほうが大切ではないか。

- 1) 手続き方言語の考え方 : 代入と再利用
- 2) 順次・選択・反復の構造化を理解させる。
- 3) 様々な手法に共通する考え方を理解させる。  
再利用と繰り返しの削減
- 4) 誤差の扱い : 誤差の発生と集積を防ぐ。

## プログラミング教育の入門コース

プログラミング教育入門コースに求められることを、森口繁一著『Excel/Basic 基礎指南』から引用する：

- (1) 自然な文脈の流れの中で、文法事項を少しずつ出していく。
- (2) 実習によって行動が要求され、それが正しければ強化が、誤りであれば修正が行われる。
- (3) 比較的短い時間で一応の達成感が得られる。
- (4) 興味と意欲次第で、さらなる発展が示唆される。

ここで(2)の「Skinner のオペラント条件付けによる学習法」は、人間ではうまくいかないことが知られている。高橋秀俊(『数理と現象』)の“人間の特性8箇条”によれば、「人間は、気まぐれである。」、「人間は怠け者である。」などと記されている。

これらの人間の特性を踏まえて設計されたものが、良いプログラミング言語の一つの条件を満たすことになる。

Excel VBAは、プログラミングにおける人間の負担を軽減する点で優れている。というのは、ワークシートという目に見える領域を中心としてプログラミングするからである。

## 結論

高校の教科『情報』と準備している大学の一般情報教育が共有すべき「情報教育」としての「プログラミング教育」についての結論は、以下の通りである。

- 1) 近い将来の学習指導要領の改訂では、『数学B』での「数値計算とアルゴリズム」に関する単元は廃止して、『情報B』に統合して、コンピュータ科学の基礎を中心とした教科『情報』とすることが望まれる。
- 2) 現在の高校学習指導要領では、アルゴリズムおよびプログラミングの取り上げ方が不十分である。
- 3) Excel VBAは、高校の『情報B』や『数学B』の授業でも、適切な補助教材があれば、十分学べるので、活用が望まれる。

## 講演要旨(2)

### 「道具性を重視したプログラミング教育」

#### はじめに

「道具(machine)」とは何であろうか。「machine」の語源「machinate」は、日本語で「たくらむ・企てる」の意味を持ち、machineの語源は、ギリシャ・ローマ時代では「力」、「企み」を意味していたが、中世ヨーロッパの劇作では「すべてを解決する『神』の登場する脚本」あるいは、「『神』を入場させる舞台装置」の意味を与えられていた。

#### 動く接面を持つ「道具」

今日では、ユーザー自身の心理的世界と対象としての物理的世界とを介在するのが、「道具・機械的世界」である。(佐伯 1988)この「道具・機械的世界」と心理的世界との境を第一接面、物理的世界との境を第二接面と呼んでいる。すると、石器時代(農耕・狩猟)に第一接面と第二接面をつないでいた道具・機械の世界は、産業革命・情報革命を経て拡大し、今日の情報社会では知的インタフェースとなることで、時間空間的に隔たった物理的世界をも対象として取り扱えるようになった。(海保・原田・黒須 1991年)このことは、心理的世界から物理的世界を操作するための道具・機械的世界は、「道具型」から認知的インタフェースとしての「秘書型」と呼ぶべきものとなったことを示唆している。ここで、「webで他人の旅行記を読んでいるとき、第一接面・第二接面は、それぞれどこか?」といった問題を考えてみよう。

表1 「動く接面」の回答例

第一接面	第二接面
キーボード・画面	コンピュータ
webブラウザ	webサーバ
コンピュータネット	作者
旅行記	旅先の土地

表1に示す回答例からわかるように、第一接面と第二接面は、ユーザーが働きかけ興味の対象によって、接面の捉え方が動いていると言える。また、Rutowski(1982)によれば、「うまくいっている」とき道具は透明性を持ち、対象世界は道具が直接作用するものであり、第二接面は道具の向こう側となるが、道具に支障があれば透明性は失われ、対象世界は道具そのものであり、第二接面は道具のこちら側である。

#### プログラミング教育

学習者がプログラミング教育で得られることは、「道具」として、プログラムが作れることや、「テクノロジー」として、システムの視点で対象を捉えられるようになることである。さらに、「リソース」として、意思決定にシステムの視点からの分析結果が使えるようになることである。

教師がプログラミング教育を行う上で注意したいことは、学習者が『プログラムの機能』に動機を持ち、本当に意図した動作か確認できるようにすることである。さらに、環境として、授業以外でも『プログラム』をできるように教育しなければならない。

<休憩> (POV-Rayによる作品提示)

#### 実践事例

次に、技術的な知識はほとんどなく“限りなく文系的な”理系とも言うべき学生(1学年40名)を対象に、「CAD/CG実習(1年夏期集中)」、「情報処理演習(2年後期:必修)」で行った実践事例を紹介する。

「CAD/CG実習」では、3次元CGのプログラミングツールとして、「POV-Rayシーン記述言語」を採用した。この言語には、「シンプルな文法」、「結果が視覚で確認できる」、「Win/Mac/Linuxなどマルチプラットフォーム」、「Freeware」と言った特徴がある。

取り上げた課題は、平行移動・回転・拡大縮小などの基本的な図形操作を必要とする図形の作成である。

「情報処理演習」では、UNIX の基礎を 90 分×3 コマ講義したあと、CGI の動作原理を理解するための課題として、web サーバを作成した。ネットワーク I/O は、tcpserver を使用し、perl で記述するが、細かな文法は、教師が書いた。

受講した学生の反応は、「十分理解し、自作できる。」、「だいたい理解し、模倣できる。」、「やりたいことはわかる、課題は友情で。」、「わからないが課題だけは出した。」、「リタイヤ」などであった。

なお、前述の「プログラミング教育で注意したいこと」に加えて、授業でプログラミングをマスターするのは、到底可能なことではなく、学習者の動機づけと自習のための能力と web 上の情報源あるいはその探し方などの情報も与えるようにすることが、大切である。

#### 質疑応答

プログラミング教育について、活発な質疑応答があった。教科「情報」の内容と「擬似言語の有効性」などが話題となった。また、プログラミング教育が、「考える」・「疑う」などの思考能力の育成に、通じるのではないかとの意見もあった。

文責：大岩幸太郎（大分大学）