

「新・試作教科書」におけるプログラミングの扱い*

兼宗 進 (一橋大学), 久野 靖 (筑波大学)

概要

高等学校普通教科「情報」は2003年に新設され、教育が進められているが、プログラミングの扱いについては、どのような内容をどの程度扱うかについての基準がなく、現在でも試行錯誤が続けられている。本報告では、「新・試作教科書」のプログラミングに関する方針を紹介し、「情報」におけるプログラミングの位置付けを検討する。

abstract

The subject “Joho” (Information) in Japanese highscools has newly started in 2003. Based on the experiences of this new subject, IPSJ committee on primary and secondary school educations has discussed on desirable curriculum of this subject on the next decade, and have written the “new experimental textbooks” for this subject. In this paper, we introduce our basic policy about programming in this textbook.

1 はじめに

高等学校普通教科「情報」[1]は2003年に新設され、教育が進められているが、プログラミングの扱いについては、どのような内容をどの程度扱うかについての基準がなく、現在も試行錯誤が続けられているのが現状である。

情報処理学会初等中等教育委員会では、来るべき改訂に向けて我々が考える「望ましい教科内容」を「新・試作教科書」[3]の形でまとめ、2006年10月に公開した。本稿では、プログラミングに関する方針を紹介し、「情報」におけるプログラミングの位置付けを検討する。

2 現行の指導要領での扱い

現行の指導要領[1]では、プログラミングの扱いは明確に規定されていない。たとえば情報Bを見ると、(2)「コンピュータの仕組みと働き」では、コンピュータの動作原理を扱っているが、「図を使った説明」や「命令がステップで動いていることを扱う程度とする」など、図を使った理解に範囲を限定している。(3)「問題のモデル化とコンピュータを活用した解決」では、現実世界の問

題をコンピュータを用いて処理することを扱っており、「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、唯一プログラミングの利用について言及されている。

多くの教科書では命令のステップごとの実行についてCPUの内部動作を解説しており、アルゴリズムについてはフローチャート等の図を用いていることが多い。モデル化とシミュレーションについても表計算ソフトを使う場合が多く、特にプログラミングについては触れられていないのが一般的である。

一方、2007年度から使われている新版の教科書では、数社からJavaScriptなどを用いたプログラミングの要素が導入され始めた。「情報」という教科におけるプログラミングの扱いは、どの程度の内容を扱い、どの場面で活用していくかについて、試行錯誤が続けられている。

3 新試作教科書での扱い

新・試作教科書では、現行課程の「情報A/B/C」の選択必修部分については、「情報B+C」の内容をおおむね含ました1つの必修科目「情報I」を設け、その上に「情報II」「情報III」を積み上げる形で選択科目を配置していくことを提案してい

*Programming explanation in New “Joho” Experimental Textbook, by Susumu KANEMUNE (Hitotsubashi University), Yasushi KUNO (University of Tsukuba)

る。それぞれの科目における、プログラミングの扱いを概観する。

3.1 情報 I

情報 I は「第 1 章 ネットワークと情報」「第 2 章 情報とその活用」「第 3 章 コンピュータと情報」「第 4 章 情報社会」という 4 部構成とした。

この中で、第 3 章においてプログラミングとそれを用いた問題解決を扱っている。図 1 に、第 3 章の内容構成案を示す。

第 3 章 コンピュータと情報
3.1 プログラミング入門
・ドリトル言語によるプログラミング体験
・命令を実行してみよう (実行すると画面が変化)
・命令を続けて書こう (命令は順に実行される)
・命令を繰り返そう (同じ命令を繰り返し実行できる)
3.2 対話的なプログラム
・ボタンを作ろう
・動作を定義しよう (押すとタートルが前進する)
・ボタンを増やそう (左右に向きを変えるボタン)
・タイマーを作ろう
・カメラにエンジンを付けよう (自動的に前進する)
・左右に動かそう (左右のボタンで操縦する)
3.3 コンピュータと手順的な自動処理
・コンピュータの構造、構成要素、CPU、メモリハードとソフト、手順的な自動処理
3.4 情報と問題解決問題とその性質、例:数式を解く問題の諸性質
・問題解決プロセスとその必要性、問題の認識/同定/記述
・問題解決プロセスの例 — 複利計算の問題、問題のモデル化、解決プロセス

図 1: 「情報 I」内容構成案の抜粋 (第 3 章)

我々は 2005 年に公開した「日本の情報教育・情報処理教育に関する提言 2005」[4]において、「手順的な自動処理」を小学校・中学校・高等学校の各段階において、すべての児童・生徒に「体験」させることを提案した。

手順的な自動処理の実践手段としては描画ソフト、音楽ソフト、表計算ソフトなど多様な切り口が考えられるが、高校段階においてはプログラミングの「体験」が自然な選択であると考えている。

ただし、現行指導要領にも「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。」と書かれているように、言語の学習が目的とならないように注意する必要がある。

そこで情報 I では、言語の説明がほとんど不要で、最初の時間からプログラミングを「体験」する目的から、教育向けオブジェクト指向言語であるドリトル [5]を採用した。

ドリトルは LOGO 言語と同様、タートルグラフィックスの機能を搭載しており、これによって生徒はごく簡単なプログラムであってもそのプログラムが動いたことの結果 (フィードバック) を視覚的な形で受け取ることができる (図 2)。

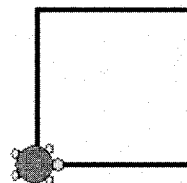


図 2: ドリトルによるタートルグラフィックス

また逆に、「このような絵を描きたい」という問題を自分で設定し、それを行うプログラムを作成するという形で問題解決を実践することができる。このように、与えられた問題ではなく、自分で設定した問題による問題解決を行うことは、モチベーションの維持のために重要である。

さらにドリトルでは、一定時間間隔で動作を行わせる機能を用いてタートルを画面上で一定ペースで動かさせ (アニメーション)、画面上にボタンを配置してそのボタンが押された時の動作を記述することでタートルの動きを対話的に制御することができ、ゲームのようなものも作ることができる (図 3)。このようなプログラムを扱うことで、生徒は日常使っているソフトウェアの動作も同様のプログラムによって作り出されていることが体験的に理解できる。

「情報 I」全体で扱うべき内容の分量にも配慮して、ドリトルによるプログラミングに割り当てる時間としては 5~8 週間程度を想定し、その中に「簡単な例題で学んだ後で自分で実際に学んだ機能を利用した作品を制作する」サイクルを 1~2 回含めるのがよいと考えている。

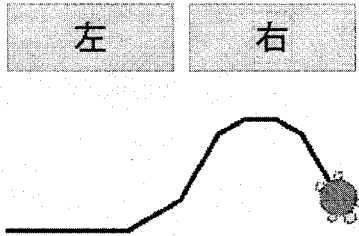


図 3: ボタンによるタートルの制御

試作教科書では、これらの学習が終わった後で、コンピュータの構造や機能、ハードウェアとソフトウェアなどについて学ぶ構成としている。これは、実際にプログラムを動かした経験に基づいてコンピュータの原理を学ぶ方が具体性があると考えたことによる。

3.2 情報 II

情報 II は「第 1 章 コンピュータとプログラミング」「第 2 章 アルゴリズムとデータ構造」「第 3 章 メディアリテラシーと情報倫理」「第 4 章 情報システムと情報社会」という 4 部構成とした。

この中で、第 1 章でプログラミングを、第 2 章でプログラミングを用いたアルゴリズムとデータ構造を扱っている。図 4 に、第 1,2 章の内容構成案を示す。

「情報 B」的な部分については、現行の「情報 B」がアルゴリズムを前面に出しているのに対し、新・試作教科書では「情報 I」で体験的にコンピュータの自動的な処理を理解させ、その土台に立って「情報 II」でアルゴリズムについて学ぶように構成している。ただしそこでも、直接アルゴリズムに入るのではなく、まず第 1 章でプログラミングや自動処理について再度入門し、「情報 I」とは別の言語でプログラムを書くことをひととおり学ぶようにしている。その課程においても、まず電卓とコンピュータの対比に基づいて「手順」を導入し、プログラミングの題材として文字を並べたお絵描きを扱うなど、手順を動かすことを重視している。第 2 章ではデータ構造とアルゴリズムを扱

第 1 章 コンピュータとプログラミング

1.1 コンピュータの動作

- ・なぜプログラミングについて学ぶのか?
- ・コンピュータと情報社会
- ・コンピュータと自動的な処理
- ・コンピュータと手順
- ・コンピュータと電卓の比較
- ・計算と手順

1.2 JavaScript によるプログラミング

- ・入力・処理・出力
- ・JavaScript 言語と HTML
- ・例題: 2 つの数の和
- ・枝分かれの記述、if 文
- ・例題: 2 つの数の大きい方を出力
- ・同じ問題に対する多様な記述の可能性

1.3 繰り返しと関数

- ・繰り返しの記述、while 文
- ・例題: 文字を用いたお絵描き
- ・関数定義とその用途
- ・計数ループと for 文
- ・例題: 関数を用いたお絵描きの改良版

第 2 章 アルゴリズムとデータ構造

2.2 アルゴリズムと段階的詳細化

- ・一般化、段階的詳細化
- ・繰り返しを含むアルゴリズム
- ・例題: 区間 2 分法による求解
- ・ループの考え方、不変条件、停止条件
- ・再帰関数の考え方
- ・例題: 階乗の再帰関数、最大公約数

2.2 データ構造とアルゴリズム

- ・データ構造、配列、例題: 合計と平均
- ・データ構造の工夫、例題: 素数の列挙、データ構造の追加による効率化 (エラストテネス)

2.3 実際の問題への適用

- ・レコード型、レコードとその用途、レコードの配列
- ・探索、線形探索のアルゴリズム、線形探索を用いたプログラム
- ・整列、単純選択法のアルゴリズム、単純選択法のプログラム

図 4: 「情報 II」内容構成案の抜粋 (第 1,2 章)

うが、人工的な例題だけでなく、実際のデータに近いものも扱うように配慮している。

3.3 プログラミングの導入

「情報 I」におけるプログラミングが「手順的な自動処理」を実地に動かしてみ、普段接しているソフトウェアとのつながりを考えさせる、という体験的色彩の強いものであるのに対し、「情報 II」では「科学的理解」の題材としてのプログラミングとアルゴリズムを扱うこととし、ある程度まで理論面および定式化を意識した形でプログラミングを扱う。このため、「情報 II」の 1 章では「情報 I」から連続という形ではなく、あらためてプログラミングを導入する形とした。

その冒頭では、情報社会の中枢に「情報を自動的に加工」する機能があることを指摘し、その本質部分にプログラムがあることをまず挙げる。

続いて、手順の概念を導入するために、具体的にイメージしやすい計算の例として「電卓による計算」をとりあげ、計算のためのキー操作を固定してデータだけ取り替えることで「同じ手順が繰り返し使える」ことを示し、ここからプログラムと手順を導入している。

続いて実際にプログラミング言語を用いた学習に進むが、ここでは言語として JavaScript を用いている。JavaScript はスクリプト言語という成り立ちから、簡潔な記述でプログラムを動かしてみることができ、また Web ブラウザに処理系が組み込まれているため、ブラウザだけで実習が行えるという利点もある。2007 年版の「情報 B」教科書では、少なくとも 2 社が JavaScript 言語を用いたプログラミングを扱っている。

「情報 I」で採用したドリトル言語で全体を統一することも考えたが、2 章であつかうアルゴリズム的な題材には JavaScript のような「伝統的な」構文の方が向いている可能性があること、また早い段階から複数のプログラム言語に接することは有益であるとの考えから、JavaScript を採用した。

扱う内容としては、まず入力と出力、枝分かれ、繰り返しなどの基本的な言語機構を学ぶが、ここで「同じ動作を実現するコードは何種類もある」

ことを取り上げておく。また、題材として「文字を並べて図形を描く」ものを採り入れることで、結果を目で確認できるように努めた。さらに、現行の教科書にはないが、関数(サブルーチン)は抽象化の手段として重要だと考え、この段階で導入するようにしている。

3.4 アルゴリズムとデータ構造

2 章「アルゴリズムとデータ構造」は現行「情報 B」の「コンピュータにおける情報の処理」を発展させた内容となっており、コンピュータサイエンスの入門部分を無理のない範囲で扱うという方針になっている。この点から、現行「情報 B」で挙げられている「アルゴリズム」だけでなく「データ構造」も並行して扱うように配慮した。

その導入部分ではアルゴリズムが持つ抽象化(一般化)の側面を取り上げ、段階的詳細化の考え方を学ぶようにした。このような方法論はやや難易度が高いが、このような考え方なしに「天下り」できあいのアルゴリズムを提示するのは望ましくないとの考えからこのようにしている。

その後、区間 2 分法による求根を題材として、実用になるアルゴリズムを体験し、また繰り返し部分における不変条件、終了条件などの考えに接し、アルゴリズムを正しく考えることを学ぶ。

さらに、やや進んだ話題として、1 章で学んだ「関数」を発展させた「再帰」についても簡単に扱っている。これまでの我々の経験から、あまり理論的な説明を前面に出さず、単に「自分で自分を呼び出すこともできる」という形で説明していけば、高校生でも十分にこの話題が消化できると考えている。

最後に、前述のように試作教科書全体として「データベース」の機能の詳細(SQL など)を扱わない代わりとして、レコード型を導入し、レコードの配列という形で現実的なデータをプログラム中で扱うことを取り上げる。その中で整理と探索という基本的なアルゴリズムを「道具として」使ってみせるようにした。

4 ヒアリングによる評価

試作した教科書の評価として、数名の教員に対してヒアリングを行った。教員のうち4名は高校「情報」の現職または授業経験があり、1名は中学校「技術・家庭科」で情報を担当している。評価は、公開された試作教科書のPDFをダウンロードしてプログラミングを扱っている章を読んでもらい、コメントを電子メールで受け取った。以下に結果を紹介する。分析はコメントに関する考察を行ったものである。人数が少ないため、特に統計的な分析は行っていない。

4.1 全体的なコメント

全体的に、教員から見て難易度が高いことがわかった。特に聞き慣れない専門用語は理解を妨げるため、わかりやすい言葉に置き換えるか、定義してから使うなど、工夫が必要である。

- 最初の試作教科書より、ずっと読みやすくなった。
- いろいろと勉強になった。
- 全体的に難しい。
 - － 厳密すぎる長文で硬い。図表やチャートを使うべき。(2名)
 - － 用語の解説が必要。「オペレーティングシステム」「プログラミング言語処理系」「フェッチ」「パイプライン方式」「オブジェクト」「カスケード」など。(2人)
- プログラムの1行ずつの意味を解説したい。リストに吹き出しを付けるなど。
- 演習問題の「～せよ」という命令口調は、高校では違和感がある。
- 小中学校と違い、高校では生徒が独習できる必要がある。

4.2 情報 I に関するコメント

オブジェクト指向の解説を工夫する必要があることがわかった。「手順的な自動処理」を定義する前に使っている箇所があった。後半のアルゴリズムの部分で前半のプログラミングを活用するなど、章としての一貫性を持たせる必要がある。

- 流れはわかりやすい。
- 最初に習うプログラムは易しいことが重要。
- 「手順的な自動処理」は定義が必要。(3名)
- オブジェクト指向の扱い。
 - － 利点の説明で必要性を伝えたい。(2名)
 - － 「もの」という例えを見直したい。表記を統一したい。(2名)
- タートルをペンと呼ぶのは紛らわしい。(2名)
- 説明が必要。代入、「!」「。」など。
- 構造化の利点を説明したい。
- アルゴリズムの節で前章のプログラミングが活かされていない。

4.3 情報 II に関するコメント

難易度が高いことがわかった。難しい理由のひとつはHTMLとJavaScriptを同時に学ぶことにあり、具体的な操作手順を示すなどサポートする必要がある。また、厳密に書きすぎて難しくならないよう留意する。

- JavaScriptはWebブラウザがあれば実行できるのでよい。
- いきなり難しくなる。入門用の言語で通したほうがわかりやすい。(2名)
- HTMLのDOCTYPEなど「おまじない」が多い。JavaScriptだけを記述したい。(2名)
- HTMLを生成する本来の使い方から入るのが自然。

- HTML で現在の時間を表示し、計算結果も表示する。
- “*” で図形を描き、表やカレンダーも表示する。
- 理解が難しい。
 - 途中からわからなくなりました。実行しながらだと理解できるかもしれない。
 - 具体的な操作手順がほしい。どの部分を入力して、どんなファイル名で保存して、どうやってプログラムを実行できるのか。
- 細部より大きな概念を伝えたい。
 - 言語の文法より、条件分岐や関数などを説明したい。
 - プログラムと数学の変数の違いを強調するより類推してもらうべき。
 - 仕様の細かい記述は理解を妨げる。
 - 疑似コードを証明する課題は不要。

5 おわりに

新・試作教科書の情報Ⅰ、情報Ⅱにおけるプログラミングの扱いを紹介し、高校教員へのヒアリングによる評価を行った。その結果、次のことが明らかになった。

プログラミングが扱われていることに対しては、特に問題は指摘されなかった。ただし、今回ヒアリングした教員は全員がプログラミングの授業経験があるため、この点については客観的な調査になっていない可能性がある。

試作教科書自体は、前回のもの [2] より改善されているという意見があった。しかし、まだ難易度が高く、十分にわかりやすい内容にはなっていない。どの章にも言えることだが、試作教科書を作成したまま、評価はほとんど行われてこなかった。今後、他の章を含めて教員や生徒による評価を行うことで、質を高めて行くことが可能である。

今回の新・試作教科書では、教育用言語である「ドリトル」と実用言語である「JavaScript」を利用して、プログラミングの基本的な概念を伝えることができた。今後は、学習者が達成感を持つような、ある程度実用的なプログラムを扱って行きたい。

情報Ⅰに関しては、プログラミングの指導に慣れていない教員が初学者に教える前提を堅持し、コンピュータの動作原理である「手順的な自動処理」を体験を通して理解する。その際、基本的な「制御構造」「オブジェクト」などの概念を体得できることが望ましい。プログラムは一度では動かないものであり、その原因を悩みながら、デバッグの体験を通して体得して行く。その意味で、最初は例題を入力して確認する学習を行いながら、最後は自分のアイデアで作品プログラムを作成する学習に発展する内容が望ましい。これはワードプロセッサやゲームなど、コンピュータ上で単体で動作するプログラムに相当する。

情報Ⅱに関しては、Web ブラウザの中で動作するプログラムの形態を維持することで、ネットワークを通して動作するプログラムの学習に発展させたい。学習する言語が JavaScript であっても、Ajax 的な学習用のライブラリを用意することで、サーバーとの通信を行い、データベースへのアクセスなどを実現することが可能である。このような体験を通して、単に画面をマウスでクリックするだけであった Web 画面の仕組みを理解でき、自発的にインターネットの世界を理解することが可能になる。

最後に、試作教科書の作成にあたって協力してくれた関係者の方々に感謝します。また、有益なコメントをくれた先生方に感謝いたします。

参考文献

- [1] 文部省, 高等学校学習指導要領解説 情報編, 開隆堂出版, 2000.
http://www.mext.go.jp/b_menu/shuppan/sonota/990301d.htm
- [2] 情報処理学会初等中等教育委員会, 高等学校普通教科『情報』試作教科書, 1998.

[http://ce.eplang.jp/index.php?
%BB%EE%BA%EE%B6%B5%B2%CA%BD%F1](http://ce.eplang.jp/index.php?%BB%EE%BA%EE%B6%B5%B2%CA%BD%F1).

- [3] 情報処理学会初等中等教育委員会, 高等学校普通教科『情報』新・試作教科書, 2006.
[http://www.ipsj.or.jp/12kyoiku/
teigen/v83joho-text0701.pdf](http://www.ipsj.or.jp/12kyoiku/teigen/v83joho-text0701.pdf)

- [4] 情報処理学会情報処理教育委員会, 日本の情報教育・情報処理教育に関する提言 2005,
[http://www.ipsj.or.jp/12kyoiku/
proposal-20051029.html](http://www.ipsj.or.jp/12kyoiku/proposal-20051029.html).

- [5] 兼宗 進, 御手洗理英, 中谷多哉子, 福井眞吾, 久野 靖, 学校教育用オブジェクト指向言語「ドリトル」の設計と実装, 情報処理学会論文誌:プログラミング, vol. 42, No. SIG 11 (PRO 12), pp. 78-90, 2001.