

2 情報教育における プログラミング利用の可能性

久野 靖 情報処理学会初等中等教育委員会／筑波大学大学院ビジネス科学研究科

はじめに

情報化社会である今日において、情報や情報技術の教育をどのように進めるかは世界のどの国においても重要な課題となっている。しかしその中であって我が国の情報教育は、(1) 割り当てられている時間数がごく少なく、(2) 情報技術に関する内容が軽視されているという点で、非常に見劣りする内容となっている。

特に(2)については、高等学校の普通教科「情報」において、(2a) 選択必修となっている3科目のうち、「情報の科学的理解」に重点を置く「情報B」の開講数が5%程度と少ないこと、(2b) その「情報B」の指導要領⁶⁾においても「アルゴリズムは学習するが、プログラミングには深入りしないこと」と注記されていることから、高校までの学校教育でプログラミングを体験している生徒の比率はきわめて小さいものとなっている。これは、他国の情報教育ではほぼ常にプログラミングの内容が含まれていることと対照的である。

筆者が所属する情報処理学会情報処理教育委員会および初等中等教育委員会では、我が国が情報技術に関してどのような問題を抱えているか、またその問題を克服するために、小学校・中学校・高等学校を通じてどのような情報教育がなされるべきかという観点から検討を進めてきた。本稿ではこれらの要点を紹介した上で、情報教育において「プログラミング」の内容を取り入れることの必要性やその具体的指針について述べる。

我が国が抱える問題

我が国では2005年から2006年にかけて、情報技術に関係した事件が連続して起こっている。情報処理教育

委員会では、これらの事件について検討し、「情報処理と情報システムの原理に対する理解の欠如」が共通の要因であることを指摘し、我が国の産業界ひいては社会全体の態度に問題があるとの警告を発している⁵⁾。

たとえば、「構造計算書偽造事件」では、ある建築士が確認申請に出した構造計算書に偽造があり、強度不足で住めない建物が大量に存在することが判明している。その要点は、構造計算書は「大臣認定プログラム」が計算出力した結果を受理する制度設計であったのに、その出力を直すことが現に可能だったことにある。もともと、デジタル情報は変更が容易であり、改ざんがないことを保証する必要があるならそのような技術を使うべきである。さらに、「結果」だけ添付して「入力データ」を添付しないという制度設計もおかしいといえる。

また、「1円61万株誤発注事件」では、ある証券会社の担当が「61万円で1株」の売り注文を出すところを間違えて「1円で61万株」の売り注文を出し、その取り消しができなかったことから多くの取り引きが成立してしまい、証券会社が膨大な損失を被っている。その要点は、東証のシステムがおかしな注文でもそのまま受け入れる方針であったこと、注文の取り消し機能にバグがあったこと、証券会社のシステムは「おかしな注文」に対する警告機能を持っていたがその警告はしょっちゅう出るので常に「OK」と返答する習慣ができていたことにある。人間は必ず間違いを犯し、ソフトには必ず間違いが含まれることを考えれば、このような設計方針が不適切なのは明らかであるし、しょっちゅう出る警告に効果がないことも専門家にとっては常識のはずである。

他の事件も含めた共通の要因として、これらのシステムを(制度まで含めて)設計した人が情報システムや情報技術の性質についてきちんと理解していなかったことがある、というのが我々の考えである。そしてこのことは

さらに、2つの側面に分けられる。

1番目の問題は、情報技術者の水準の低さである。もちろん、すべての情報技術者の水準が低いというつもりはないし、優れた技術者が多数いることも分かっている。しかし、上記のように問題のあるシステムが現に作られ、事件が頻発した以上、それらを作った技術者に問題がなかったとは言えないはずである。

今日の我が国では、実際のシステム開発作業は中国・インド等にアウトソースできるから、多数の情報技術者は必要ない、という議論に接することがある。しかしアウトソースするとしても、システムの仕様を決めて発注し納品物を検収することは必要である。しかし、前記のような事件を見ていると、実際にその任にあたっている技術者の水準に問題があると考へざるを得ない。

2番目の問題は、国民全体の情報および情報技術に対する理解や関心の不足である。このうち重要なのは後者の「関心」である…というのは、「関心」がないなら、学んで理解しようとする意思も働かないからである。

しかるに、現在の我が国の人たちは、それ自体を仕事とする情報技術の専門家以外は、庶民から企業のトップに至るまで、情報技術の中身には関心がないように見える。その典型的な態度は「情報技術は大切かもしれないが、自分は情報技術のことは分からないから、専門家に任せる」というものである。

しかし、情報技術や情報システムが自分にとって大きな影響を及ぼす以上、それに関心を持ち、理解しようとするのは本来、各自の個人としての責務であるはずである。それを放棄して「丸投げ」されて作られたシステムであれば、担当した情報技術者がいかに優秀であっても、発注側の意を十分に汲んだものにはなり得ない。

情報教育が目標とすべきこと

前章で述べたことがらに基づき、我々としては我が国の情報教育が次の2点を目標とすべきであると考え、

- (1) 国民全体の「情報水準」の底上げ
- (2) 関心を持つ人材への学習機会の提供

これまで(1)のような目標は、「全員に対する最低水準の保証」のように捉えられることが多かったが、そのような均一的な考えは望ましくない。個人の進路や職業はさまざまであり、それぞれの進路に応じた水準の情報や情報技術に対する理解を持ってもらうことを指して「情報水準」の底上げと呼んでいる。たとえば経営者として企業を率いていく人材に対して「すべての国民と同様

の情報技術に対する最低水準の理解」では不十分なのは明らかである。

一方、(2)については、これまで我が国の情報教育では取り上げられてこなかった部分だといえる。しかし、情報技術に関する関心や適性は個人ごとに大きく違いがあり、その中から特に適性を持つ人材を見出して育成することは、中国やインドをはじめ多くの国で行われていることである。我が国では英才教育的活動は難しい面があるが、少なくとも各個人が自分の持つ情報技術に対する適性を見出せるような機会を提供することは、本人にとっても社会にとっても大きな利益となるはずである。

なお、文部科学省は初等中等教育全体を通じた情報教育の目標を「情報社会を生きる力の育成」と位置付け、その具体的な内容として「情報活用の実践力」「情報の科学的理解」「情報社会に参画する態度」の3要素を掲げている。我々としては、これらの目標は上記(1)の内容を「さまざまな分野の方向」に具体化したものに相当しており、それを「さまざまな個人に対する適用性の方向」にも広げることで上記2目標が達成されると考えている。

上記2目標と初等中等教育以外の段階における実現のための方策などについては、文献4)において詳しく議論している。

プログラミングの位置付け

我々は、前章で挙げた情報教育の2目標を達成する上で、プログラミングは有効な手段の1つとなると考えている。しかしそれについて議論する前に、現在の情報教育におけるプログラミングの位置付けについて整理したい。

冒頭でも述べたように、今日の普通教科「情報」において、プログラミングを取り上げている例はきわめて限られている。また、情報教育にプログラミングを導入することに対しては、強い「拒絶反応」があることが多い(そのことが今日の状況につながっているといえる)。プログラミングの導入に反対する論拠として、次のようなものが挙げられてきた。

- (a) コンピュータを使うのにプログラミングは不必要である。
- (b) 国民全員がプログラマになる必要はないではないか。
- (c) プログラミングは難しすぎて教えるのが困難である。
- (d) プログラミングを教えるだけの時間的余裕がない。
- (e) 「情報」教員の多くはプログラミングを教える力がない。

(a), (b)については共にその通りであるが、我々は「コンピュータを使うのに必要だから」「国民全員をプログラマにしたいから」プログラミングを教えようと主張しているわけではない。プログラミングを体験することが、児童／生徒の情報および情報処理の原理理解に資するところ大だから、学習の手段としてこれを扱いたいというのが我々の主張である。

(c)については、BASICなど既存の言語で教育を行っている経験を持つ教員などに多く見られる意見である。しかし、BASICは30年以上前に作られた言語であり、予約語が英語であるなど入門教育に使用する上で障害となる側面を多く持っている。今日では、教育により適した特性を持つ言語の研究が多く進められており、これらの新しい言語を用いることで、児童／生徒により効果的にプログラミングを体験させられる。

また、(a)～(c)を通じて、「自力で一定のプログラムを書けるようにさせる」という目標が暗に前提となっていることも指摘したい。我々としては、興味や適性を持つ児童／生徒がその段階まで進むことはもちろん望むところであるが、全員を対象とする場合は「プログラムを動かしてみる」「プログラムを直すと対応してその動作も変化する体験を持つ」ことを目指せばよいと考える。この程度であれば、必要とする時間数は数時間程度であり、現状でも十分実現できるものと考えている（もちろん、「情報」の教員が研修などを通じてより高い専門性を身につけていくことも必要なことは間違いない）。

なぜプログラミングを教えるのか

色々準備(?)に手間がかかったが、ここで「なぜ情報教育においてプログラミングを扱いたいのか」について、我々が考える理由を3つ挙げさせていただく。

理由1：コンピュータの構造／原理／特性を学ぶにはプログラミングを「体験」することが必要

たとえば、小学校の「理科」の授業で全員が試験管を持って「でんぶんのヨード反応」を体験するのは全員を化学研究者や技術者にするためだろうか？ もちろんそうではなく、「物質どうしが反応することでさまざまな現象が起きる」体験を持たせることが重要だからこのような教育課程が組まれているものと考えられる。

「情報」についても同様のことがいえる。たとえば、次のようなことがらは教員が講義しても身につけにくい、自らプログラムを動かして体験してみれば強く実感され

ることである。

- ソフトウェアは、綿密に計画して注意深く作っても、必ずしも思った通りの動作をするようにはならない。
- ソフトウェアが思った通りに動作しない場合、実際には人間が間違いを犯している（人間はどんなに注意しても必ず間違いを犯す）。
- ソフトウェア自体が間違っていなくても、与えるデータが正しくなければ、間違った答えが得られてしまう。
- ソフトウェアはほんの1カ所を変更しただけでもまったく動作が違ってしまうことがある。実行が永遠に止まらなくなることも珍しくない。

これらのことがらを「体験として」実感しておくことは、その児童／生徒のコンピュータや情報システムに対する適切な態度をかたち作る上で有効なはずである。

なお、現在の教科「情報」では、コンピュータの原理として「CPUは命令を1個ずつ取り出して実行してゆく」のような内容を教えることが想定されている。確かにこれは間違いではないが、命令を1個ずつ実行するという動作は現実のソフトウェアの挙動からかなり隔たっているため、このことを学んでも現実のコンピュータや情報システムに対する理解にはつながりにくい。それよりは、同じ時間数でも「プログラムを打ち込んで動かしてみる体験」の方が上記の理由からずっと効果的であると予想される。

理由2：コンピュータや情報システムに対して「こうしたらどうなるか」が予測できるようになる

理由1では体験の内容として「プログラムを打ち込み動かしてみる」「プログラムを直すと対応してその動作も変化する」ことを挙げたが、授業で取り上げる以上、その対象となるプログラムの動作内容についても一通り説明するはずである。その場合、一定数の児童／生徒については、そのコードを「読んで理解できる」ことも期待できる。これが可能だった場合、次のようなことがらが実感されるはずである。

- プログラム内部のデータは目に見えないが、それがプログラムの動作によって画面などに現れて見えることになる。
- プログラムの実行はきわめて高速であるが、処理するステップ数が多くなればそれに呼応して時間もかかる。

- プログラムが何かを表示する場合、それは単にプログラムにそのような処理が記述されていたからであり、それ以上でもそれ以下でもない（にもかかわらず、そのプログラムに対面する人間の方で何かの意味があるかのように見てしまうことがある）。

これらの理解を持つことは、理由1よりやや深いレベルでコンピュータや情報システムの挙動に対する理解を持つことになり、「このように操作したらどのようなか」をある程度まで予測できるようにする効果を持つと考える。

また、この段階の理解を持つておくことで、冒頭に挙げた構造計算書事件や1円61万株事件のような問題が「起きてしまう前に気づく」能力も養えるのではと期待している。

理由3：関心と適性を持つ児童／生徒に対し初等中等教育段階でそのことを見出す機会を与える

プログラミングやソフトウェアの作成には他のことと違う特性が多くあるため、その方面に関心や適性を持つ児童／生徒を見出すには、実際にプログラムを扱わせてみるのがほぼ唯一の方法である。

実際に我々の経験でも、ドリトル¹⁾などの教育用言語を学んだことがきっかけで、従来の教科の学習では目立たなかった児童が自分の適性に気づき、もっとプログラミングを学びたいと意思表示するようになった事例がある。

児童／生徒には一人ごとにさまざまな適性や得意分野があり、その中から自分が適したものを見出すことによって将来自分が生活の糧を得、また社会に貢献する方向を見つけていくことは、教育の重要な役割である。初等中等教育段階で全員にプログラムを動かす体験をさせることは、この方面に適性や関心を持つ生徒を発掘し、その方面の才能を伸ばすきっかけを与えるものとして不可欠であると考ええる。

また、このことと関連して、適性／関心を持つことが分かった生徒に対しては、選択科目のようなかたちでより進んだ系統的な学習の機会を提供することで、我が国の将来を担うIT人材や、IT分野に進まないとしても、IT分野について一定の理解を持つ人材を育成していくことも必要である。

まとめ

ここまでで、我々の主張として「初等中等段階でプログラムのようなものを体験する」ことを提案してきた。その段階や分量については文献4)でも述べているが、小学校・中学校・高等学校の各段階において、数時間程度ずつの「経験」を持ってもらうことが当面適切であると考えている。

またそこでは、前述のように「プログラム」という形態に対する強い抵抗があることもまったく無視するわけにはいかないと考える。そのため、プログラムに限らず「表計算ソフトウェアの自動計算機能」「音楽ソフトウェアの自動演奏機能」「描画ソフトウェアの自動描画機能」などを例示し、これら「一定の規則に従って自動的に処理が進んでいくもの」を体験させることも有効であると指摘している。

プログラミングを含む高校普通教科「情報」の具体的なカリキュラム構成や題材の提案については、「試作教科書」という形でまとめたものを提案・公表している^{2), 3)}。これについてはあくまでも一案であり、今後も情報教育やそこにおけるプログラミングの活用についてさらに検討していきたいと考えている。

参考文献

- 1) 兼宗 進, 御手洗理英, 中谷多哉子, 福井真吾, 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 (PRO12), pp.78-90 (2001).
- 2) 久野, 兼宗, 西田, 山之上, 小井土, 神沼, 辰己, 奥村, 長, 中野, 並木, 和田: 教科「情報」新・試作教科書の提案, 情報処理学会高校教科「情報」シンポジウム 2006 資料集, pp.60-196 (2006).
- 3) 情報処理学会初等中等教育委員会: 高校普通教科「情報」新・試作教科書 2006.12.11 版 (2006), <http://www.ipsj.or.jp/12kyoiku/teigen/v83joho-text0612.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>
- 5) 情報処理学会情報処理教育委員会: 2005年後半から2006年初頭にかけての事件と情報教育の関連に関するコメント, <http://www.ipsj.or.jp/12kyoiku/statement2006.html>
- 6) 文部省: 高等学校学習指導要領解説 情報編, 開隆堂出版 (2000). (平成19年3月26日受付)

久野 靖 (正会員)

kuno@gssm.otsuka.tsukuba.ac.jp

1984年東京工業大学大学院博士後期課程単位取得退学。同年同大理学部情報科学科助手。1989年筑波大学大学院講師。助教授を経て現在、同大学院ビジネス科学研究科教授。理学博士。