

小学校教員養成課程におけるプログラミング教育

Programming Education in Elementary School Teacher Training Courses

南葉 宗弘^{*1}, 宮寺 庸造^{*1}, 樋山 淳雄^{*1}, 北澤 武^{*1}, 加藤 直樹^{*2}
Munehiro Namba^{*1}, Youzou Miyadera^{*2}, Atsuo Hazezama, Takeshi Kitazawa, Naoki Kato

^{*1}東京学芸大学 技術・情報科学講座

^{*1}Department of Tech. and Informatics, Tokyo Gakugei University

^{*2}東京学芸大学 教育実践研究支援センター

^{*2}Center for the Research and Support of Educational Practice, Tokyo Gakugei University

あらまし：東京学芸大学では平成31年のカリキュラム改訂にあたり，初等教育教員養成課程における共通選択必修科目の中に「小学校におけるプログラミング教育」と題した授業を開設する．この授業科目の開設に向けたプロジェクトチームにおいて議論された“小学校教員のためのプログラミング教育”の内容を紹介するとともに，すべての小学校教員が身につけるべき計算論的思考（あるいはプログラミング的思考）に関わる知識・技能を提案する．

キーワード：初等教育，プログラミング教育，教員養成，情報教育

1. はじめに

新しい学習指導要領において小学校段階におけるプログラミング学習が盛り込まれた．これに呼応するように，さまざまな企業・団体が子供たちを対象としたプログラミング教育に関する本を出版したり，子供たちが直接プログラミングを体験するワークショップや教室を開催したり，子供たちがゲーム感覚で楽しみながら取り組むことのできるプログラミング教材を提供したりするなど，その事実だけに目を向ければ今はまさにブームの様相を呈している．しかしその一方で，実施をせまられている小学校においては，現職の教員が必ずしもプログラミングに関する知識や経験を持つわけではなく，実際に何をどのように教えるのかについて苦慮している．当然，小学校教員を志す者に対して，その養成段階からプログラミングに関わる基礎的な理論や学習活動における教育実践上の考え方等を身に付けさせることが強く求められている．

東京学芸大学では平成31年度の改訂カリキュラムから，初等教育教員養成課程における選択必修科目の中に「小学校プログラミング教育」と題した授業科目を開設する．本稿では，この授業科目の開設に向けたプロジェクトチームにおいて議論された“小学校教員のためのプログラミング教育”の内容を紹介するとともに，全ての小学校教員が身に付けるべき計算論的思考（あるいはプログラミング的思考）に関わる知識・技能を提案する．

2. プログラミング時の思考過程

諸外国では，プログラミングを含む情報科学教育の目的について，問題をコンピュータに行わせるまでの分析，解決方法の提案，その実施と評価等の，コンピュータサイエンティストが用いていると考えられている一連の思考方法（計算論的思考）や知識・技術（コンピュータサイエンス）などを学習するこ

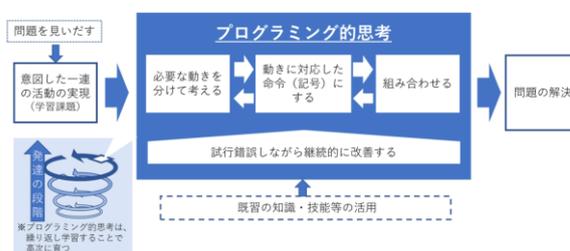


図1 プログラミング的思考

(文部科学省「小学校プログラミング教育の手引き(第一版)」13ページの図4を引用)

とであるとしている⁽¹⁾．あくまでプログラミングはその中の一部であり，プログラミングを通じて，上述のことを学ぶという位置付けである．おそらくこれに準じて，我が国の新しい学習指導要領でも，プログラミングを通じて，コンピュータに意図した処理を行わせるために必要な論理的思考力（プログラミング的思考）を身につけさせることを目的として掲げたものと考えられる⁽²⁾．アルゴリズムの考え方やその表記の仕方等，より高度な内容は中学校や高校で学習することになっているため，小学校では，コンピュータに何かをさせるには手順が必要なのだということに“気付かせる”ことが大事だという趣旨であろう．したがって，図1に示すような①必要な動きを考える，②動きを命令（記号）に置き換える，③命令（記号）をどのように組み合わせれば自分の意図した動きになるかを考える，という3ステップを試行錯誤しながら繰り返す，という過程こそが涵養すべき論理的思考力であると説明している⁽³⁾．

しかし我々は開設予定の授業科目の内容を議論する中で，このようなプログラミング時の思考過程の説明は不十分であり，特にゼロからプログラミング及びその教育について学ぼうとする初学者に対しては誤解を与えかねないとの結論に至った．コンピュ

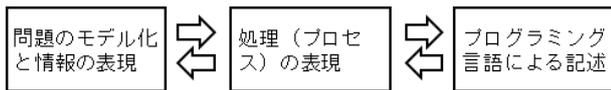


図2 プログラムを創る思考過程

一々に意図した処理を行わせるために必要な論理的思考力とは、意図する活動を実現するような命令の組み合わせを見いだす能力であると勘違いされてしまうと、先の図1は思考過程を表したものというよりむしろ実作業、すなわちコーディング作業過程を表したものと理解されてしまうだろう。そのまま素直に授業が行われると「〇〇というプログラミングツールの使い方だけを学んだだけで、何の意義があるのか分からなかった」といった疑問を抱いたプログラミング嫌いの子供たちを増やす原因となりかねない⁽⁴⁾。

3. プログラムを創るために必要な力

プログラムを創るという観点だけに限定したとしても、その思考過程には一貫してアブストラクション (abstraction: 抽象化) が必要不可欠である^{(5) (6)}。そこで我々は小学校教員が理解しておきたい“プログラムを創る思考過程”とは図2に示すような①問題のモデル化と情報の表現、②処理(プロセス)の表現、③プログラミング言語による記述、の3ステップの試行錯誤的な繰り返しであると主張したい。

3.1 問題のモデル化と情報の表現

問題の解決につながるようなプログラムを考えるにあたっては、まずはその問題自体をコンピュータ上で表現しなくてはならない。問題の解決に必要な情報だけを抽出し、それらをデータとして数や記号を用いて表すことが必要である。

3.2 処理(プロセス)の表現

どのように処理を行っていくかを考えるにあたっては、まずプログラムの開始時の状態と、終了時の状態を、一つ目のステップで定義したデータを用いて表現しなくてはならない。その後、どのように加工を施していけば開始時の状態を終了時の状態に変化させられるかを考える。この時、よほど単純な問題でない限り、細かな加工処理を順番に筋道立てて組み合わせて考えるようなことはすべきではない。全体として終了時の状態まで持って行ける処理を複数の小さな処理に分割したり、処理の中に処理を包含させたりする等して、意図した結果を導く手続きを抽象的に考える⁽⁶⁾ことが肝要である。

3.3 プログラミング言語による記述

二つ目のステップで考えた処理を実際にコンピュータに行わせるには、プログラミング言語が規定する記号を用いて表現しなくてはならない。一般にはこの作業をコーディングと呼んでいる。特定のプログラミング言語を学ぶことは小学校における教育目的ではないし、そうあるべきでもない。しかし、ど

のような命令が使えるのか、どのような文法が許されているかが、一つ目と二つ目の両ステップで考える表現に大きく影響するという事実には十分注意する必要がある。

加えて、例えば数学において数学記号を用いて数と数の関係を表現するように、あるいは物理学において法則という記号表現を用いて自然現象を表現するように、コンピュータ上での処理はプログラミング言語という記号で表現するのだということを認識させ、将来的に、その抽象世界で考えをめぐらせることができるようになる素地を作ることも大事なのではないかと考える。

4. 小学校教員のためのプログラミング教育

未来を担う子供たちにとって“コンピュータが「魔法の箱」ではなく、より主体的に活用する⁽³⁾”ことができるようになるには、小学校段階におけるプログラミング教育が“面白かった”というだけの体験に終わってはならないし、意義の分かりづらいコーディング作業に終始してはならない。もちろん、プログラミング教育を拡大解釈し、コンピュータを抜きにしたプログラミング論を展開するようなことがあってはならない。子供たちには、身のまわりの物事をデータとして見る力を養い、それらを自らの目的のためにコンピュータを用いて処理させることができることに気付き、その思考過程を身に付けることのできるようになる教育がなされるべきである。

そのためには教員養成課程においてコンピュータの仕組みを学び、その上で、プログラムを創る思考過程が理解できるようになる授業科目をカリキュラムの中に適切に設けなくてはならないと考える。

参考文献

- (1) 太田剛, 森本容介, 加藤 浩: “諸外国のプログラミング教育を含む情報教育カリキュラムに関する調査—英国, オーストラリア, 米国を中心として—”, 日本教育工学会論文誌, 第40巻, 第3号, pp. 197-208 (2016)
- (2) 文部科学省: 小学校学習指導要領・解説 (2017) http://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm (2018年5月アクセス確認)
- (3) 文部科学省: 小学校プログラミング教育の手引き(第一版) (2018) http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm (2018年5月アクセス確認)
- (4) 鶴飼佑: “英国でのコンピュータサイエンス教育の今とは? 必修化3年後の状況と日本のこれからの「プログラミング教育」を考える”, NPO 法人 CANVAS ワークショップ (2017年12月14日) <http://canvas.ws/workshop/16595> (2018年5月アクセス確認)
- (5) Jeannette M. Wing: “Computational Thinking: What and Why?”, Link Magazine (2010) <http://www.cs.cmu.edu/~CompThink/papers.html> (2018年5月アクセス確認)
- (6) Randall W. Jensen: “Structured Programming”, IEEE Computer, vol. 14, no. 3, pp. 31-48 (1981)