

プログラミングで教える情報の科学的な理解

— 国立教育政策研究所教育課程研究指定事業（共通教科情報）の研究実践より —

鎌田 高德

文部科学省は次期学習指導要領において小・中・高において「プログラミング教育の必修化」を決めた。その背景にあるのは、経済産業省が発表した「IT人材の最新動向と将来推計に関する調査結果」のデータであり、2020年には36.9万人、2030年には2030年78.9万人のIT人材が不足すると報告されている。そのため2020年より、まずは小学校から段階的にプログラミング教育が始められることとなった。

そんな中、筆者は勤務校において平成28年・29度の2年間、国立教育政策研究所教育課程研究指定校事業（共通教科情報）の指定を受けた。その研究指定の中で新学習指導要領を意識し、プログラミングを通して情報の科学的な理解を深める研究実践を行ってきた。本稿では、これからの教科情報においてプログラミング“で”何を身につけさせるのか、それが情報の科学的な理解であると筆者は考え、実際に行ってきた授業実践の1つを報告する。

1. はじめに

高校では2022年より次期学習指導要領が始まり、「情報Ⅰ」と「情報Ⅱ」という新科目が設定される。その新科目では情報の科学的な理解がより重視され、また「プログラミング」のキーワードが入った単元が追加される予定である。

こうしたプログラミング教育の流れを受けて、「プログラミング教育で大切なのは、プログラムの書き方を覚えることではなく、問題解

決には必要な手順があり、その1つ1つの手順の意図を理解し、思ったとおりに操る思考力を育てることである」（黒上・堀田2017）と指摘している。プログラミングはあくまでも目標を達成するための手段の一つであり、目的ではない。コンピュータやネットワークの動作の仕組みをプログラムの手順に置き換え、その意図を思った通りにプログラミングすることが、プログラミング教育のねらいとなるのではないだろうか。

プログラミング教育を行う上で大切なのは、プログラミング“を”教えるのか、プログラミング“で”教えるのか、この2つの視点をバランスよく考えた上でプログラミング教育を進めることである。表1は本稿の教科「社会と情報」において、プログラミングを取り入れた2016年度の授業実践の一覧である。これらの授業実践は全て、プログラミングを覚えることを目的としているのではなく、基数変換の仕組みや、暗号の解読の手順をプログラミングの手順に置き換えて、その仕組みを体験的に理解することを目的としている。例えば②の題材は、プログラミングで「公開鍵暗号方式の解読困難性」を教えている（大石2017）。また、③の題材は、プログラミングで「ネットワークではIPアドレスやホスト名で接続先を識別していること」を教えている（小松2017）。本稿では、筆者が考えた①の題材で、プログラミング“で”「基数変換の手順をプログラミングの手順に置き換えて、仕組みの理解を深めた」実践報告を行う。

表1 プログラミングを取り入れた授業一覧

プログラミングを取り入れた題材	プログラミング導入のねらい
①基数変換を行うプログラムを制作しよう (言語:ドリトル)	基数変換の手順をプログラミングに置き換えて考えさせる
②公開鍵暗号方式の解読の手順を考え、その手順をプログラムで実行しよう (言語:VBA)	公開鍵の素因数分解の手順をプログラムに実行させ、なぜ公開鍵が簡単に解読されないか検証させる
③クライアントサーバシステムの役割をプログラミングで体験しよう (言語:ドリトル)	サーバ役とクライアント役に役割を分けて、その仕組みを体験的に学ばせる

2. プログラミングで情報の科学的な理解を深める

2.1. 情報の科学的な理解とは

文部科学省は情報教育の目標として(1)情報活用の実践力,(2)情報の科学的な理解,(3)情報社会に参加する態度,の3つの観点で整理している。ここでの情報の科学的な理解とは「情報活用の基礎となる情報手段の特性の理解と、情報を適切に扱ったり、自らの情報活用を評価・改善するための基礎的な理論や方法の理解」としている。

これから知識基盤社会の急速な進展により、情報の科学的な理解に裏打ちされた情報活用能力を身に付けることがより一層重要になってきている。情報の科学的な理解を深めることは、優れた情報活用の実践力につながる。また情報社会が果たしている役割を科学的に理解することにより、情報社会に参画するより良い態度を身に付けることにつながる。プログラミング教育と情報教育の目標の親和性を考えた時に、自分で考えた情報の科学的な仕組みを手順化し、

その手順をプログラミング“で”表現することで、体験的に情報の科学的な理解を深める活動は非常に重要になってくると考えている。

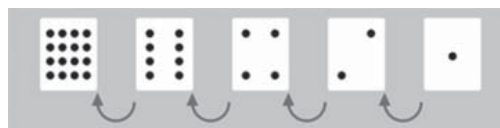
2.2. 基数変換を行うプログラミングを制作しよう

プログラミングを通して情報の科学的な理解を深めさせることをねらいとして、2進数から10進数への変換を行うプログラミングの授業を行った。プログラミングを取り入れたのは、自分たちで考えた基数変換の仕組みの手順を、プログラミングの手順に置き換えさせることで、情報の科学的な理解を深めることを目的としたからである。

2.3. アンブラグド点を数える

コンピュータの原理である2進数「0」と「1」のデジタル信号のデータ処理の仕組みに対する効果的な学習方法として、コンピュータサイエンスアンブラグドがある(兼宗ら2008)。2進法の仕組みについて、図1に示すアンブラグドの「点を数える」の2進数カード(点16までのカード)を用意し、生徒らにどのようにカードを活用しながらどんな数字を表現できるか考えさせ、2進法の仕組みについて学ばせた。

図1 アンブラグド「点を数える」



2.4. 手計算で2進数から10進数へ

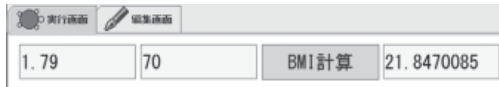
次に手計算で2進数がどのように10進数に変換されるか手計算の手順を説明し、生徒らに2進数から10進数への変換を手計算させた。

2.5. ドリトルで簡単な計算機を作成

次にプログラミング教育用ソフトウェアドリ

トルで簡単な計算 ($X + Y = Z$) のプログラミングを行った。四則演算を計算する方法について一通り学び、最後に演習問題として図2に示すようなBMIの計算プログラムを作らせた。ここでプログラミング“を”学んでいないと、プログラミングの手順に置き換える時に、どのような計算式を立てればいいのかイメージがでない。

図2 BMIの計算プログラム



2.6. プログラミングの手順に置き換える

ドリトルで簡単な計算をさせる力を身につかせた後で、生徒らに図3に示すような2ビットまでの計算に対応していたプログラムを生徒に配布した。配布後、プログラムの動作確認を行った後で、図4に示す3ビットまでの計算に対応したプログラムに書き換えさせた。

図3 2ビットの計算プログラムを配布し動作確認

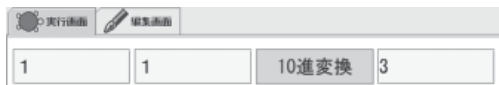


図4 3ビットの計算プログラムに書き換えさせる



2.7. プログラムをテストする

2ビットから3ビットへのプログラムを書き換えただけでは、自分の考えたプログラミングの手順が正しいか分からない。手順の確認のために表2に示す相互チェック表を用意し、プログラミングの手順が正しいか実際に2進数の数値を入力させ、プログラムのテストを通して手順が正しいかの試行錯誤を行わせた。テストの

結果が正しかったのか、隣の人とペアを作り、お互いにしっかりとできているか相互チェックをさせた。

表2 プログラムのテストと相互チェック表

2進数	10進数の結果	相互チェック
010	2	
101	5	
110	6	

2.8. デジタル信号を文字に変換させる

テストが上手くいったら、対応ビット数を増やしたプログラムを作らせた。生徒らが一通りプログラムを完成したところで、デジタル信号として送られてきた2進数を10進数に変換し、その10進数とASCIIコード表を参照しながら文字に変換させ、デジタル信号を文字に変換させる活動を行った。

普段、自分たちが使用しているスマートフォンなどの機器が、受信したデジタル信号をプログラムでどのように変換し文字として表現されているか、自分で作ったプログラムで体験させることで情報の科学的な理解を深めさせた。

3. 結果

3.1 生徒に実施したアンケートデータ

プログラミングで基数変換の仕組みが理解できたかアンケートを実施した。79%の生徒がプログラミングを使って基数変換の仕組みを理解することができたと答えている。

表3 アンケートデータ（総回答数141）

質問	回答数
プログラミングを使わなくても理解できた	10%
プログラミングを使って理解できた	79%
理解できなかった	11%

3.2 プログラムによるデバッグテスト

基数変換のプログラムを行えた次の授業の導入で、前回の振り返りとして計算式にエラーがあるプログラムを配布し、修正できるかデバッグのテストを実施した。表4に示す結果にあるように73%の生徒がプログラミングのエラーを見つけ修正することができた。アンケートでは89%の生徒が理解できたとしても、プログラミングのデバッグを行うと73%となり、16%の生徒がしっかりと理解できていなかったことが分かった。

表4 デバッグのテスト結果 (総回答数147)

質 問	回答数
プログラミングを修正できた	73%
プログラミングを修正できなかった	27%

4. 考察

アンケートとデバッグのテストの結果より、おおむねの生徒がプログラミングを通して基数変換の仕組みを理解することができたといえる。

プログラミング教育において大切なのは、プログラミングで“で”何を教えるかである。プログラミング“を”教えることを目的としない授業設計が非常に重要になる。また、これからプログラミング教育の中で、生徒の学習活動をどう評価するのか、まだ何も見えてきていない。今回の実践で扱ったように、デバッグによるテストも一つの有効な評価の手立ての1つであると感している。

【参考文献】

著Tim Bell/Ian, 翻訳 兼宗進(2008) コンピュータを使わない情報教育アンプラグドコンピュータサイエンス,

教育用プログラミング言語「ドリトル」

<http://dolittle.eplang.jp>

黒上晴夫, 堀田龍也 (2017) プログラミング教育導入前に知っておきたい思考のアイデア, 小学館

大石智広 (2017), 暗号を解読せよプログラミングで学ぶ公開鍵暗号, 第10回全国高等学校情報教育研究大会 (東京大会) 記念冊子, p28-29

小松一智 (2017), これからを見据えた「プログラミングで教える」をテーマにしたプログラミング実践, 第10回全国高等学校情報教育研究大会 (東京大会) 記念冊子, p60-61