

Vol. 48

## CONTENTS

【コラム】日本技術士会と情報処理学会の連携について… 黒澤 兵夫

【解説】情報科教員のための教員免許更新講習（後）… 久野 靖

【解説】はじめてのプログラミングとつまずき… 岡本 雅子

## COLUMN



### 日本技術士会と情報処理学会の連携について



技術士は科学技術に関する技術的専門知識と高等の応用能力および豊富な実務経験を有し、公益を確保するため高い技術者倫理を備えた優れた技術者の育成を図ることを目的とした国家資格である。技術士は21の技術部門を広くカバーしているが、このうち情報分野に対応しているのが情報工学部門である。2014年には最高裁から地裁・高裁の長官に対し、専門訴訟の際には技術士を使うようにとの通達も出され、技術士の活動範囲も広がっている。

2001年に技術士制度は国際整合するエンジニア資格(PE, Professional Engineer)として技術者倫理、資質向上(CPD, Continuing Professional Development)等について大幅に改定され、工学教育との連携(JABEE)も強化された。しかし、技術士制度の社会的認知が十分でないため、2014年12月末の時点で情報工学部門の技術士は1,950名、2014年度技術士第二次試験における情報工学部門合格者は85名(JABEE認定プログラム修了者3名)にとどまっている。社会における情報系専門職人材の重要性が増す中で、より多くのJABEE認定プログラム修了者が技術士を志すことが望まれる。

日本技術士会は、2014年8月に情報処理学会との間で「高度な情報技術者の育成・可視化および情報系プロフェッショナルコミュニティの形成に関する覚書」を締結した。覚書に基づき行われている主な連携を以下に示す。このうち、①項と②項については、技術士法第45条(守秘義務)を遵守しつつ業務を遂行している。

#### ① CITP への協力(企業認定, 個人認証)

情報処理学会が開始した認定情報技術者制度(CITP, Certified IT Professional)に技術士会も審査員や制度検討委員会の委員として協力している。企業認定は、社内で独自の資格制度を持つ大手SIerを主な対象とするアクレディテーションの仕組みである。個人認証は、技術者個人が情報処理学会に直接申請し審査するもので、中小企業やフリーランスの技術者の利用を想定している。

#### ② 技術士試験委員

情報処理学会へ情報工学部門の試験委員などを依頼している。

#### ③ CPD コラボレーション

情報処理学会へ技術士・情報工学部門のCPD企画の一部を依頼している。将来的にはCITPのCPDとの相互連携も図る予定である。

現在、技術士・情報工学と情報処理技術者試験(高度試験)合格者の相互活用に関する検討が、科学技術・学術審議会技術士分科会制度検討特別委員会相互活用WGで行われている。これを推進するために、情報処理学会が策定したカリキュラム標準J07を参照して基礎・CS, CE, SE, IS, IT(情報基盤)について技術士一次試験および二次試験に対応するBOK(知識体系, Body of Knowledge)の案がとりまとめられた。技術士・情報工学, 情報処理技術者試験・高度試験, CITPの間で資格制度の相互活用が図られるようになれば、日本における情報系プロフェッショナルコミュニティが形成され、世界に向けた積極的な発信が可能になることが期待される。

黒澤兵夫(TAKE 国際技術士研究所, 技術士(情報工学・総合技術監理))

LOGOTYPE DESIGN...Megumi Nakata, ILLUSTRATION&PAGE LAYOUT DESIGN...Miyu Kuno

# 情報科教員のための 教員免許更新講習（後）

久野 靖

(SSR:情報処理学会「会員の力を社会につなげる」研究グループ/筑波大学)

## 前回までのあらすじ

我々（本会 SSR「会員の力を社会につなげる」研究グループおよび教員免許更新 WG）は、情報科教員にとって有益な教員免許更新講習の実現を目指して、2012年・2013年夏に実施した研修「大学の授業を聞こう」に併設する形で教員免許更新講習を実施することとした。実施は2014年8月4日～6日の3日間であり、1日目は「情報社会と情報倫理の現状」の内容で実施した。

## 2日目：プログラミング教育の考え方

2日目のスケジュールを表-1に示す。この日は細かく演習を行い、講義は久野、演習解説と討論司会は補助講師が担当する構成とした（演習時は講師全員で見まわり助言）。

まずアルゴリズムとプログラムの説明の後、すぐ Ruby 言語の入門の内容を講義し、引き続いて短い例題を題材に実習環境で Ruby プログラムを動かす演習を行った。次にコンピュータでの数値の表現や計算の誤差について説明し、計算誤差を確認する演習を行った。さらに続いて制御構造の解説に進み、枝分かれについて説明したあと、「2数の大きい方を返す」「3数の大きい方を返す」などの問題による演習を行った。午前の最後は繰り返しを説明し、題材として数値積分のコードを説明したあと、誤差に対する配慮が必要となる注意点を取り上げてコードの改良について説明した。

午後はまず、午前中の最後の演習2件について、中西が演習の解説を行った。そのあと、制御構造の組合せについての解説があり、Fizzbuzz問題などを題材とした演習を行った。続いて、データ構造を簡単に説明し、配列とその利用方法について説明した。時間が不足してきたため、この部分の演習は行わず、間辺が制御構造の組合せと合わせて演習解説を行った。最後に、画像のデータ構造をプログラム内で用意してデータを書き込み、画像ファイルに出力する例題を説明した上で、実際に画像ファイルを生成して確認し、さらにプログラムを変更して自分

時間帯	内容
9:30～10:00	アルゴリズムとプログラム / Ruby
10:00～10:15	演習：プログラムを動かす
10:15～10:30	数値の表現と計算の誤差
10:30～10:45	演習：計算の誤差の確認
10:45～11:15	制御構造の位置付け、枝分かれ
11:15～11:30	演習：枝分かれの記述
11:30～12:00	繰り返しの記述、数値積分
13:00～13:45	枝分かれ・繰り返しの演習解説
13:45～14:00	制御構造の組合せとその意義
14:00～14:15	演習：制御構造の組合せの記述
14:15～14:30	データ構造、配列とその利用
14:30～15:00	制御構造の組合せ演習解説
15:00～15:15	2次元配列による画像の表現
15:15～15:45	演習：画像を生成するプログラム
15:45～16:15	グループによる意見交換と集約
16:30～17:00	更新講習：試験、SSR：グループの意見の紹介と討論

表-1 講習2日目のスケジュール

時間帯	内容
9:30～10:30	情報学の参照基準とその考え方
10:30～11:00	東京大学のメディア関連講義
11:00～11:30	討論：参照基準と高校の情報教育
12:30～13:30	情報（科）学の考え方
13:30～14:00	計算の機構、コンピュータの内部構造
14:00～15:00	IC トレーナによる論理回路演習
15:15～16:15	討論：情報（科）学と高校情報科の教育
16:30～17:00	更新講習：試験、SSR：全体を通しての意見と討論

表-2 講習3日目のスケジュール

の計画した画像を出力する演習を行った。

最後は演習解説は行わず、間辺の司会により、その場でグループを構成して学習内容のまとめを話し合わせる時間とした。その後、更新講習受講者は前日同様別室で試験となり、SSR 参加者はその間にグループの話し合い内容を紹介した後、自由討論とした。議論の内容としては次のものがあつた。

- プログラミングに関する質問+普段疑問に思っていること
- どの言語を使用するか
- 早く終わってしまった生徒への対応方法

この日の冒頭に参加者に挙手で尋ねたところ、プログラミング未経験者は3分の1くらい含まれていたが、それらの人も含めて全員が問題レベルの難易はあつても、それぞれ自分に合った課題をやってみることができた。この点では、このような「実際に大学で教えられているやり方で学んでみる」体験は、プログラミングの技能を身に付けるという点も含めて、教員の研修内容として有用なのではないかと感じた。

### 3日目：情報科学の考え方

3日目のスケジュールを表-2に示す。この日の午前中はすべて萩谷が担当した。まず、情報学の参照基準<sup>1)</sup>の説明があり、情報科の親学問としての情報学の定義、文系から理系にまでまたがる広がりや、その特徴的な内容などが解説された。引き続い

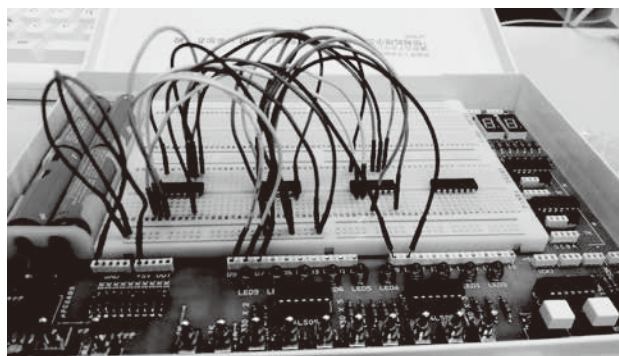


図-1 IC トレーナで配線を行ったようす

て、文系の情報学とはどのようなものかという話題に移り、文系の情報学の例として、東京大学のメディア関連の講義としてどのようなものがあり、どのような内容であるかについて紹介がなされた。その後討論に移り、参照基準が高校の情報科とどのようにかかわるのか等の議論がなされた。

続いて午後も萩谷による講義から始まり、まず「情報(科)学の考え方」として、チューリングマシンなど主要な計算モデルとその意味の説明があり、続いて「計算の機構」としてコンピュータのしくみ、論理回路などの説明があつた。これらは実際に東京大学で実施されている授業の抜粋であつた。

引き続き、実習として東京大学の学部1年生が必修で受講する科目「情報」の中に含まれるIC トレーナによる論理回路の実習を実施した。この実習と引き続き討論は、滑川と長嶋が担当した。IC トレーナはケースに入ったブレッドボード内にAND/OR/XOR/NANDの各ゲートが(4つずつ)入ったIC4個と、周辺に電源(乾電池)、スイッチ、LEDが用意されていて、ジャンパ線でこれらを結んで回路を構成するものである(図-1)。実習内容としては、NANDゲートの動作確認、NANDを1つ以上用いて他の論理ゲートを実装する、半加算機/全加算機を実装する、などの課題を各自で実施した。

実習修了後、まとめの時間として今回の内容全体に対する討論を行ったが、午前中の内容は午前中の討論で扱っており、また直前のIC トレーナの実習が印象が強かつたためか、IC トレーナに関する議



区分	8/4	8/5	8/6
免許講習受講者	9	6	6
一般受講者	29	22	22
合計	38	28	28

表-3 情報処理学会免許更新講習 2014 受講者数

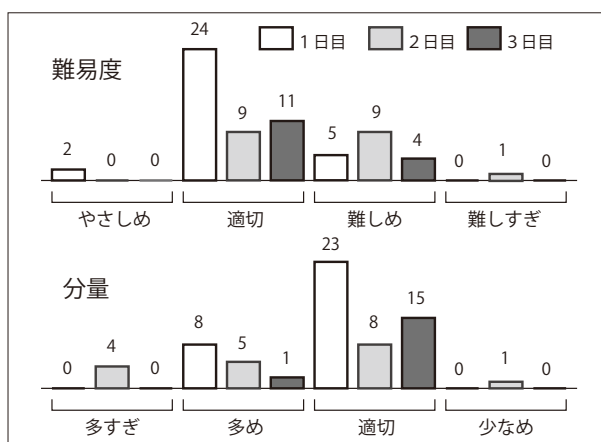


図-2 受講者アンケートの「難易度」「分量」の結果

論が中心となった。議論の内容としては次のものがあった。

- 回路実習をすることの教育的な価値について。
- IC トレーナは大変分かりやすく熱中した。高校でもやりたい。
- 電気は見えないことから難しい面があり、失敗もあるのでは(失敗も必要だという意見もあった)。
- IC トレーナの回路と実際のコンピュータのギャップについて。

これからも、IC トレーナによる回路の体験は、「現実のものを触ってもらおう」ことにより、受講者にとって大きなインパクトをもたらしたことが分かる。この後は更新講習の受講者は試験となり、それと並行して SSR 側では 3 日間の講習全体についての意見交換や、主任講師 3 名による振り返りなどがあった。

## 受講数とアンケート結果

今回の受講者数を表-3 に示す。免許更新講習を今年度から開始し、しかも申請期日の関係で広報を開始できたのが 5 月からであったことを考慮すれば、免許講習受講者は満足すべき人数だったと考え

る(アンケートでも前年度中のアナウンスをとの意見があった)。これに SSR 側の参加者を加えると、人数的には各日とも前年までの研修並みであった。

今回は、各日とも同一内容の簡単なアンケートを実施した。その中から「難易度」「分量」の項目を図-2 に示す(いずれも 5 段階の質問だが「やさしすぎ」「少なすぎ」は 0 だった)。これによれば、難易度は適切ないし難しめ、分量は適切ないし多めという意見が多い。2 日目については、プログラミングの実習をやったということで、プログラミングの初学者は難しいという意見が多くなる一方、既修得者は高度な部分まで実施してほしいと感じたように思える。

表-4, 5 に自由記述の抜粋を示す。実習指向や実際に授業で役立つ内容に対する志向が強いことが分かるが、一方で 3 日目の情報科学の概観のような内容も好まれたことが分かる。

## 今後に向けて

今回の教員免許講習は、初めての試みとしては、受講者の満足も得られ、比較的うまく運営できたのではないかと考えている。その一方で、倫理とモラル、高校の教員による学校設備や授業例、IC トレーナによる実習などの、高校で使えるような内容が比較的歓迎され、プログラミングや情報科学など大学の内容を体験してもらおうとする内容は「難しい」という評価を得ることが多かった。これに対しては、コース内容を明確に提示し、納得してもらった上で受講できるようにしたい。

本事業は来年度以降も継続していく予定であり、また WG 内では、夏だけでなく年度末に受講の需要が増えることに対応すべきという議論もされている。これからも「生徒によりよく教えられる情報科教員のためのサポート」を旗印に活動を続けていきたい。

1 日目
<ul style="list-style-type: none"> <li>・ 実習をもう少し入れると良いと思いました</li> <li>・ なるべく演習方式の授業がありがたいです</li> <li>・ 実際に高等学校や中等教育で実践されている先生方の発表を聞いてよかった</li> <li>・ 専門分野ではないので筆記試験が苦しかった</li> <li>・ 「問題解決の学習方法について」がとてもよかった</li> <li>・ ジレンマおよび学校教育の情報化が特に勉強になりました</li> <li>・ 自分の苦手分野だったので、いくらか考えを補強できた</li> <li>・ いろいろと考えさせられる部分があった</li> <li>・ 情報倫理に関して、一方的な講義になりがちだが、いい授業作りのヒントをいただけたと思います</li> <li>・ 講義だけでなく実習を交えた盛りだくさんの研修会だと思います。事前に資料をいただければ少しよかったと思いました</li> <li>・ 言葉1つ1つ、しっかりと把握して深い理解にたつことが教員に課せられていると思いました</li> <li>・ 教員免許講習と一緒に、お話できる先生の幅が広がった。研究協議は、更新講習受講者の方々は試験のため不参加だったが、できれば一緒に協議できる時間があるとよかった。</li> </ul>
2 日目
<ul style="list-style-type: none"> <li>・ 資料が盛りだくさんだった。open question はいい内容</li> <li>・ ディスカッションの時間も、もう少し欲しい</li> <li>・ Ruby を体験できとても参考になりました</li> <li>・ 専門的過ぎて分かりにくかったです。もっと1つ1つの言葉をかみ砕いて説明してほしいです。私自身の勉強不足もありますが…</li> <li>・ プログラミングの題材がよかった</li> <li>・ なかなか難しかったです。プログラムを学ぶ意義を考えると大切であると思いました</li> <li>・ 内容が充実してよかった。特に最後の意見交換は内容が濃く、参考になりました</li> <li>・ 配列以降についてもっと時間をかけてほしい。演習説明部分は難易度により時間をかけてほしい</li> <li>・ とても勉強になりました。もっと多くの方が参加されるとよいのと思います</li> </ul>

表-4 意見の自由記述（抜粋）

3 日目
<ul style="list-style-type: none"> <li>・ 今日の内容は本来は情報系学科の専門的な内容にあたると思うが、東大生は全員が履修するとのことで、教養の高さに驚かされる</li> <li>・ 当日の講義内容変更は非常にHOTな話題を取り込む意味で良いが、願わくば資料のハードコピーが欲しかった</li> <li>・ 「情報科学の広がり」について萩谷教授より講義をうけ最新の情報を聞くことができ、大きな収穫となりました</li> <li>・ ICトレーナの実習はとても難しかったですが面白かったです。自分の知識がもう少しあればもっとはまっていけるのではと感じた</li> <li>・ 午前中の講義はもう少しコンパクトにまとめ、その分実習に時間をかけた方がよい。ICトレーナの実習はとてもよかったです</li> <li>・ 最後の討議が楽しかった</li> <li>・ ICトレーナを初めてやりました。とても興味を持てるものだと感じましたが、授業で説明するのは難しいと考えました</li> <li>・ 3日間受講しましたが、今日のお話が一番興味を持ってました。「情報学」とは？ とか「情報学の在り方」をこれからの自分の問題としてよく考えていきたいと思います</li> <li>・ 知識だけでなく、幅広い雑学（ネタ）を地道に探究していかななくてはならないと思いました</li> </ul>

表-5 意見の自由記述（続き）

#### 参考文献

- 1) 萩谷昌己：情報学を定義する—情報学分野の参照基準，情報処理，Vol.55, No.7, pp.734-743, <http://www.ipsj.or.jp/magazine/jyohosanshokijyun.html> (July 2014).

(2014年11月1日受付)

久野 靖（正会員） [kuno@gssm.otsuka.tsukuba.ac.jp](mailto:kuno@gssm.otsuka.tsukuba.ac.jp)

1984年東京工業大学理工学研究科情報科学専攻単位取得退学。同年同大理学部情報科学科助手。筑波大学講師、助教授を経て現在、同大学ビジネスサイエンス系教授。理学博士。プログラミング言語、ユーザインタフェース、情報教育に関心を持つ。



# はじめてのプログラミングとつまずき

岡本雅子

京都大学高等教育研究開発推進センター

## 学習のスタイル

コンピュータの高性能化と一般社会への普及は、小中学校におけるプログラミング教育の実施が検討されるなど、学習層の拡大にまで及んでいます。それだけではありません。学習スタイルをも大きく変えています。たとえば、かつて高等教育機関では、「FORTRAN」や「C 言語」などの言語が教えられていましたが、ここでは、教員が講義形態で文法や各命令の機能について説明する、いわゆる座学の授業が中心でした。これにはプログラムを実行させて結果を得るまでに長い待ち時間を要するなど機能的な制約が大きかったからです。ただし、独学でプログラミングを始めた方の中には、様子が異なるという方もいらっしゃるかもしれません。たとえば、1980 年代の「マイコンブーム」の中で趣味としてプログラミングを始めた方の場合、MSX 規格の PC など、「BASIC」の入門機を介してプログラミングに触れている場合が少なくないと思いますが、これらの PC を含め家庭用 PC で使用された BASIC は、インタプリタ経由ですぐに実行されるため、演習形式で学ぶことが可能でした。現在、プログラミングのテキスト教材は数多く出版されています。これらを見ると、どれをとっても、同じようにサンプルプログラムを模倣して記述し、そして実行し、その実行結果を確認するといった過程を中心に学習するように編集されています。現代のプログラミング学習もまた同様に、こうした過程を中心に実施されている場合が多いものと考えられます。

本稿では、このようにサンプルプログラムを模倣して実行し、結果を確認しながら学んでいく演習過程を特に「写経型学習過程」と呼称し、とりわけこの過程における初学者のつまずきについて述べていきたいと思っています。

## 熱力学方程式って…何ですか？

先日、知人の K さんから「プログラミングを学びたいのでアドバイスが欲しい」との相談を受けました。詳しく話を聞くと、実はこの K さんはこれまでに何度か学習を試みて計 3 回も挫折した経験があるそうです。それらの経験は、初学者の「写経型学習」におけるつまずきを説明する上でちょうど良いと思いますので、K さんの同意を得た上で、ご紹介します。

### □ 1 回目の挫折(Kさん 11 歳)

K さんは、クリスマスに、日本語 BASIC を搭載したパーソナルコンピュータ『びゅう太』をプレゼントしてもらいました。初めて記述したプログラムは「オト 1 オン」。付属の説明書にあった最も短い命令文だそうです。キーボードからの入力を終わると、「Return Key」を押して実行です。キーを押すと、「ピッ」と音がしますが、ただそれだけで何も始まりません。オト 1 オンは「ピッ」という電子音を発生させるプログラムですから当然です。しかし、少年は、「ピッ」という音をプログラム実行を知らせる音で、その後でほかに何か音が出力されると思いこん

でいたため、このプログラムの実行に失敗したと考えました。何度やっても同じ結果になります。Kさんはここでプログラミングを学習する意欲を失います。そのプログラムが「ピッ」という音を発生させるためにあること、そして、当時、実行に成功していたこと、それに気付いたのは、ずっと後のことだそうです。

## □ 2 回目の挫折(Kさん 14 歳)

中学生になったKさんは、MSX で BASIC にチャレンジします。当時、オーディオカセットテープを記憶媒体として使用することが一般的で、そのためには、BASIC を使用する必要がありました。配線し、カセットプレーヤをセットした後、命令を記述して実行です。しかし、テープは回りません。Kさんには、この一連の作業のどこに問題があるのか、それが、分かりませんでした。ここでまた、BASIC の使用を諦めます。

## □ 3 回目の挫折(Kさん 19 歳)

大学生になったKさんは、選択科目として「FORTRAN」を用いたプログラミングの授業を履修しました。この授業では、熱力学方程式をプログラミングすることが目標に据えられていました。それまで物理学を履修していなかったKさんは熱力学方程式と言われてもいったい何のことか分かりません。結果として、「プログラミング以前の問題として、何をしたいのかさっぱり分からなかった」ため、3 回目の授業以降、出席をやめてしまいました。

## つまずきの類型

先述のKさんは、どうやらすぐに匙を投げってしまう方ようです。しかしながら、私はKさんに対して、「プログラミング技術を身に付けるためには多少の根気が必要です」などとは申し上げておりません。なぜなら、学習科学の視点から見ると、これらのつまずきはすべて、学習環境にその多くの原因があるからです。

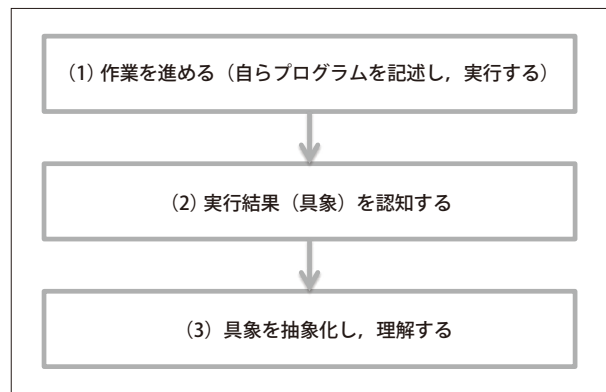


図-1 写経型学習の学習過程

写経型学習は、図-1 のとおり、3 つの段階を介して学ぶ過程です。Kさんの2 回目の挫折は、手順が分かりにくかったことに起因していて、(1) の段階におけるつまずきです。よく見られる事例を挙げると、たとえば、「if 文」が学習課題であったとしても、その課題自体の理解ではなく、プログラムのタイプミスに起因する文法エラーなどの修正にかかわるつまずきによって学習が滞ってしまう場合がありますが、これも同様の過程で見られるつまずきです。また、Kさんが実行結果の「音」を認知できなかった1 回目の挫折は、(2) の段階におけるつまずきの事例に該当します。これらのように、当該学習課題を学ぶために必要とされる先行知識の不足を原因とするつまずきは、学習一般において広く見られるもので、「認知的負荷理論 (Cognitive Load Theory)」においては、「非本質的な認知的負荷 (Extraneous or Ineffective Cognitive Load)」の1 つとして類型化されています。

一方、しばしば、「for 文」などの理解における困難性が指摘されていますが、このように学習課題それ自体が複雑であるなどの理由で理解が困難な場合には、「本質的な認知的負荷 (Intrinsic Cognitive Load)」に起因するつまずきの発生が想定されます。

これらを学習過程に沿ってつまずきを類型化<sup>3)</sup>すると表-1 のとおりつまずきの全体像が見えてきます。なお、元になった類型は、C 言語の初級プログラマ養成研修を実施している企業を対象に調査を行い、写経型学習過程で発生するつまずきを抽出してまとめたものです。





表-1 プログラミングの「写経型学習過程」におけるつまずきの類型

	本質的な認知的負荷を伴う 学習過程によるつまずき	非本質的な認知的負荷を伴う 学習過程によるつまずき
写経型学習を遂行する上で 自立的に作業することが できない場合	/	<b>【要因】</b> 学習支援環境の不備 <b>【事例】</b> 実行するための命令「a.exe」が分からない →「実行結果を確認できない」
		<b>【要因】</b> 学習支援環境の不備 <b>【事例】</b> 「expected `;` before `return`」という「return」という命令の前に文法エラー（syntax error）があるという表示から該当箇所を発見することができない →「エラー発見時に修正が必要な箇所を発見することができないのでエラーの修正ができず実行結果を確認できない」
写経型学習の過程から目的とする内容を学びとることができない場合	<b>【要因】</b> 複雑性あるいは類似の概念を学んだことがないという学習者にとっての概念の新規性 <b>【事例】</b> 条件分岐処理において、論理演算子を用いて記述する場合と入れ子で記述する場合を混同している	<b>【要因】</b> 学習支援環境の不備 <b>【事例】</b> 実行結果に小数点が表示されない（作成したプログラムでは、変数の型と出力の形式が異なることについてテキスト内に説明がなかった） →「実際に入力したプログラムの内容と画面に表示される実行結果の関連付けができず、何を学んでいるのか理解できない」

## ■ 類型に応じた改善策

これまで、つまずきを類型してきましたが、さらに、これら3つの類型にかかわるつまずきの原因や対応する学習方略について指摘あるいは提案されている研究をそれぞれ紹介していきたいと思えます。

### (1) 「作業の自立性」にかかわるつまずきと対応する学習方略 (図-1「(1) 作業を進める (自らプログラムを記述し、実行する)」に該当)

学習者がプログラミングを学ぶ際、コンパイルの手順やエラーの修正などは、特に初学者がつまずくところです。筆者は、こうしたつまずきの排除を主な目的としたC言語の初学者向けプログラミング教材を開発して実践で使用したところ、手順にかかわるつまずきが大幅に減少し、それに伴ってプログラムの概念そのものについても理解が大きく促進されたという結果を得ています。同教材では、「コンパイル手順やエラー発見の方法に関する手順」に関してテキスト本体とは別の冊子にまとめ、学習者が逐次参照しながら作業を行えるようにしたほか、同冊子に掲載するエラーの事例を増加させるなどしています。

### (2) 「作業を介した理解」にかかわるつまずきにおいて具体例の認知の困難性を伴う場合 (図-1「(2) 実行結果(具象)を認知する」に該当)

写経型学習では、記述(命令)が示す動作を確認する過程においてその学習目的を達成しようとするならば、動作は記述を反映したもものとして認識されるよう明示されているほうが良いでしょう。最初に述べましたようにKさんのつまずきの1つは、プログラムの実行結果を認知できなかったことが原因でした。同事例のように、実際に学習に使用されているサンプルプログラムの中には、当該命令の実行が動作として顕在化していないために、動作を確認しただけでは当該命令との対応関係を認識できないものが散見されます。こうした視点から、筆者は、動作を認知する際の推察や判別の要素をできるだけ排除するため、「視覚的顕在化」(表-2)という概念に従って、非本質的な認知的負荷の排除を試みた教材を開発し、同教材(図-2)を用いた実践でその効果を確認しております。

### (3) 「作業を介した理解」にかかわるつまずきにおいて当該学習課題それ自体の認知の困難性を伴う場



視覚的顕在化の4つの方法	改善前の問題点	視覚的に顕在な状態	視覚的顕在化による具体的解決法
可視化	表示が小さい 動作が瞬時に実行されてしまう	1) 大きさ、速さなどにおいて視認可能な動作であること（視認性）	表示を大きくする 1つ1つの動作を視認可能な速さで実行されるようにする
識別容易化	動作がほかの表示の中に埋没してしまい認識しにくい	2) 周囲の視覚的要素と区別して認識できること（判別性）	動作のあらわれる場所を分離する 他の視覚的要素から際立たせる
予測可能化	どの場所でのような動作が実行されているか分かりにくい	3) 視認の主体が予期するあるいは容易に予測することのできる場所で予測する動作が実行される（予測可能性）	実行される場所や動作をあらかじめ明示する 既存の知識や経験をもとに容易に予期できる動作の種類、動作のあらわれる場所に変更する
分離化	複数の処理の結果が区別しにくい一連の動作としてあらわれる	4) ほかの命令に基づく動作と区別、分離して視認できること（独立性）	当該処理に対応する動作を1つに絞る 複数の動作をそれぞれ区別が可能なかたちに分離する

表-2 プログラミング学習における視覚的顕在化

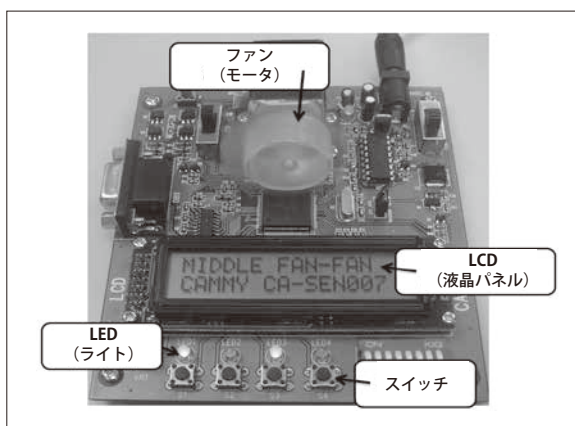


図-2 開発した教材

合(図-1「(3)具象を抽象化し、理解する」に該当)

ここでは、当該学習課題そのものの認知の困難性に伴うつまずきについての報告事例あるいはそれに対応した学習方略について紹介したいと思います。

たとえば、for文の学習について、「類似概念をこれまでに目にしたことがない」こと、「多くのデータから判断を下す、かなり複雑な問題」であり、「認知的負荷が大きい」ことがその主な要因ではないかとの指摘があります。このような場合について、河内谷<sup>2)</sup>は、類似概念が既習でない場合について、当該概念を構成している複数の要素を分解して提示し、個々の要素に類似の概念を照らし合わせて説明することで理解に導くことができるのではないかと提案しています。また、長谷川らは、初学者を対象とした授業の中で、制御構造を表す図を提示し、処理の流れのイメージが形成されることによってプログラ

ミングの理解を支援することができ、学習効果が得られることを明らかにしています<sup>1)</sup>。

## つまずき個所はほかにもたくさん

Kさんのつまずき個所のように非本質的な認知的負荷を伴う学習過程によるつまずきはほかにもたくさんあります。これまで、「つまずき個所」の事例からプログラミングの教材やカリキュラムを作成する際に工夫すべきいくつかの要素について説明してきました。本稿では、ページ数の制約からやや簡略化しております。興味を持たれた方がいらっしゃいましたら、岡本ら<sup>3)</sup>の論文も合わせてお読みいただけたら幸いです。

### 参考文献

- 1) 長谷川聡, 山住富也: プログラミング教育と学習者のイメージ形成(その2), 名古屋文理短期大学紀要, (23), pp.9-14 (1997).
- 2) 河内谷幸子: プログラミング言語の学習法— for文の理解に関する認知心理学的分析—, 言語と文化, 法政大学言語・文化センター編, 3, pp.19-35 (2006).
- 3) 岡本雅子, 喜多一: プログラミングの「写経型学習」における初学者のつまずきの類型化とその考察, バイデリア, 滋賀大学教育実践研究指導センター紀要, (22), pp.49-53 (2014). (2015年2月1日受付)

※本稿は、岡本ら<sup>3)</sup>の一部について再構成したものです。

岡本雅子 (正会員) okamoto.masako.8v@kyoto-u.ac.jp

京都大学高等教育研究開発推進センター特定助教。民間企業でシステムエンジニアとして勤務後、名古屋大学大学院国際開発研究科を経て、京都大学大学院情報学研究科で博士(情報学)を取得。専門は情報教育。

