

# 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ひます。

## 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ひますので、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」。附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ひこん

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。

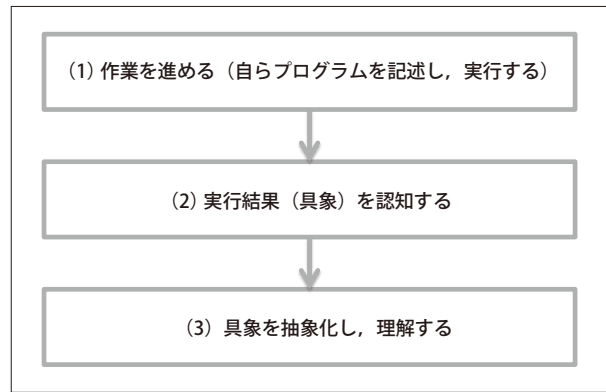


図-1 写経型学習の学習過程

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います。

### (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

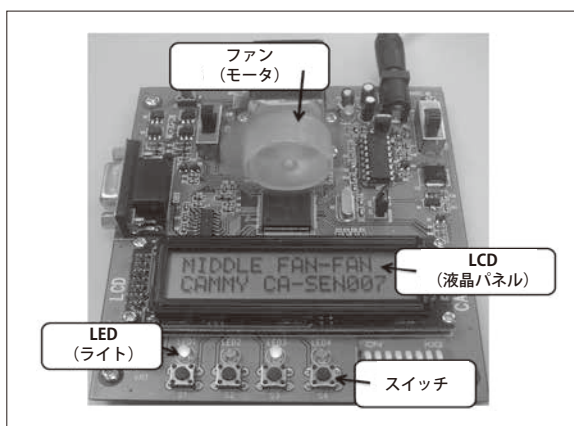


図-2 開発した教材

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。

