

# 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

## A Study of Novices Missteps in *Shakyo*-Style Learning” of Computer Programming

岡本 雅子<sup>1,2</sup>  
Masako OKAMOTO

喜多 一<sup>3</sup>  
Hajime KITA

1 滋賀大学教育学部 2 京都大学大学院情報学研究科 3 京都大学国際高等教育院

1 Faculty of Education, Shiga University 2 Graduate School of Informatics, Kyoto University

3 Institute for Liberal Arts and Sciences, Kyoto University

**<要約>** プログラミング学習において、初学者はまず、文法などに係る概念的な説明を読んだり、聞いたりして宣言的知識を獲得した後、サンプルプログラムを模倣するなどの基礎的演習過程（本稿では「写経型学習過程」と呼称する）を経て、実際に活用できる知識、すなわち「プログラミングスキル」を身に付けていく。本稿では、初学者に見られる学習の困難性（潜在的なつまずきの要因）を2つに類型化するとともに、これに従って、「写経型学習過程」を対象とした認知科学的アプローチについて概説し、認知科学的視点から考えうる学習方略について検討する。

**<キーワード>** プログラミング, 教材デザイン, 写経型学習, 初学者, 認知的負荷理論

**< Keywords >** Computer Programming, Course Design, *Shakyo*-Style Learning, Novices, Cognitive Load Theory

### 1. はじめに

コンピュータプログラミングの学習では、一般に、教員あるいは教材が提示したサンプルプログラムを学習者が模倣して入力、実行し、その結果を確認する過程（本稿では、その他の実習や演習と弁別するため「写経型学習過程（岡本 2013b）」と呼称する）を経る。ほとんどの初学者を対象としたテキスト教材（例えば、柴田 2004, 高橋 2012）においてそうであるように、プログラミングに係る個々の文法や命令を学んだ後で、実際にプログラムを打ち込んで動作させるといった作業が要求される。単にそれらの知識、つまり、認知科学で定義されているところの「宣言的知識 (Declarative Knowledge) <sup>1</sup>」を記憶することだけが目的であれば、「実際にプログラムを打ち込んで動作を確認させる」必要はないが、プログラミング学習における最も重要な目的が、個々の宣言的知識を記憶することではなく、それらを整理、統合して有機的に結び付け、利用可能な知識として習得することであることを考えれば、「写経型学習過程」が必要であり、

重要であることは明らかである。

しかしながら、プログラミングの学習においては、一般に、写経型学習過程に入ったばかりの段階で、初学者にさまざまなつまずきが見受けられ、それが学習を困難にしている。本稿では、この段階の初学者のつまずきを認知的負荷理論に基づいて類型化するとともに、その類型に即して学習を促進するための方略について考察する。

### 2. 写経型学習過程における初学者のつまずき

プログラミング学習では、一般に、写経型学習過程に入ったばかりの段階で、プログラムのタイプミスに起因する文法エラーが多発する。そのため、学習者はエラー修正の対応に追われることが多いが、これは学習者にとって今まで経験したことのない過程であるため、容易な作業ではない。例えば、コンパイル（作成したプログラムをコンピュータが実行可能な形式に変換する）操作の際、ほとんどのプログラミング言語の開発環境では、タイプミスに起因するメッセージは専門用語を含む英文で表示されるが、これを手掛かりにエラーを発見するという作業そのものの方略を知らなければ対処できない。

また、うまく実行出来たとしても、「プログラムの実行結果が想定していた結果と異なっていた」あるいは「目的とする実行結果が得られていない」ということもある。このように、実行はできたが所期の動作が

---

1 「宣言的知識」とは、Andersonによると「事実に関わる知識 (knowing that) であり、意識的に利用可能な形で保持されるものである」（Anderson 1985）とされる。例えば、宣言的知識の獲得とは、「応仁の乱は西暦 1467 年に起こった」とだけ記憶することである。

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化

	本質的な認知的負荷 (intrinsic cognitive load) を伴う学習過程によるつまずき	非本質的な認知的負荷 (extraneous or ineffective cognitive load) を伴う学習過程によるつまずき
「作業の自立性」に係るつまずき		<p>【潜在的なつまずきの要因】 教材などの学習支援環境の不備 【事例】 実行するための命令「a.exe」がわからない ⇒「実行結果を確認できない」</p> <p>【潜在的なつまずきの要因】 教材などの学習支援環境の不備 【事例】 「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 ⇒「エラー発生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」</p>
「作業を介した理解」に係るつまずき		<p>【潜在的なつまずきの要因】 教材などの学習支援環境の不備 【事例】 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」</p>

得られなかったというような場合には、目的に対して間違った記述でも、文法上の誤記がないため、当然ながらエラーメッセージが表示されない。この場合、学習者が厳密に結果を再現しなければならないという意識をもっていなければ、誤ったことにすら気付かないことがあるだけでなく、気付いた場合でも、修正を必要とする箇所を探す手掛かりが示されていないので、文法上のエラー修正に劣らず困難性を伴う。

### 3. 認知的負荷理論から見た初学者のつまずき

上述したような文法エラーや実行時のエラー箇所を自分で発見できずデバッグ（プログラムの誤りを探して修正する作業）ができないなどのつまずきについては、江木・竹内（2009）、そして太田（2009）などによって指摘されている。このように当該学習課題そのものではなく、それ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) <sup>2)</sup>」(Sweller et. al., 1998, Pass et. al., 2003) においては、「非本質的な認知的負荷 (extraneous or ineffective cognitive load)」のひとつとして類型化されている。

写経型学習過程は、提示されたプログラムの実行それ自体ではなく、その過程を通してプログラムの基本構造を理解することが主たる目的である。したがっ

て、滞りなくコンパイルを済ませて目的とする実行結果を確認できたとしても、それで学習目標を達成したことにはならない。具体例による学習は、具体例を実行を通してその背後にある基本構造を理解するとともに、同様の具体例を自ら構成できるようにすることを求めるもので認知科学的視点から見ると、「宣言的知識を利用可能な知識へと変換することを促進する」(大島 2008) とされている。写経型学習過程とは、すなわち、宣言的知識を定着させ、知識スキーマを構築していく過程に該当するわけである。しかしながら、この過程を通して学習課題の基本的な要求事項それ自体の理解に至らない場合も少なくない。例えば、著者らは、写経型学習過程を進める中で「何が何だかわからぬまま、指示された通りのことをこなしていた」(岡本ら 2013a) とするアンケート結果を得ている。また、その要因等については、場合によって異なるためここでは触れないが、写経型学習をともなう実践報告において、「順次処理（プログラムの命令を上から順番に

2 「認知的負荷理論」とは、「新たな知識や技能をより効果的に学習できるようにその学習がもたらす認知負荷のあり方を考えるもの」(Sweller et. al., 1998, Pass et. al., 2003) で、「認知負荷」とは、学習者がある課題を処理する際にどれほどの認知処理容量を割り当てる必要があるのかを指す概念である。

実行すること)を理解することでさえも初学者には難しい場合がある」(岡本ら 2010) ほか、「for 文の入れ子的な利用も初学者が典型的に困難さを抱えるものである」(阿部 2009) などの指摘がある。これらは、不適切な教材を使用することなどによる「非本質的な認知的負荷」に起因する場合や、学習課題自体の複雑性などの「本質的な認知的負荷 (intrinsic cognitive load)」に起因する場合が想定される。

ここで、「写経型学習」過程におけるつまずきについて提示した類型(岡本ら 2013b)を一部改変、および新たに認知科学的視点を追記して表 1 のとおり示したい。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ある。

表 1 では、写経型学習に係わるつまずきを「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」と「写経型学習の過程から目的とする内容を取り取ることができない場合」の 2 つに類型化した。さらに、本稿では、先行知識の学習に及ぼす負荷を認知的負荷理論を用いて (1) 本質的な認知的負荷を伴う学習過程によるつまずき、(2) 非本質的な認知的負荷を伴う学習過程によるつまずきの 2 つに区別した。前者は、学習課題を構成する情報の間に見られる関係性の複雑さと、その課題に対する学習者の熟達化の程度によって決まるものとされる。後者は、実際に学習内容を理解していくためには本質的には関係ない活動を強いることによって学習者の中に生じるものであり、学習課題それ自体の理解とは無関係で、学習にとって余計な負荷だといえる。

#### 4. つまずきの各類型に対応する学習方略

前章では、著者らがプログラミング教育の場で調査したつまずきを類型化して示したが、本章では、これら 3 つの類型に係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する。

##### 4.1. 「作業の自立性」に係るつまずきと対応する学習方略

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解決方法については、先行知識としてもっているに越したことはないが、必要に応じて学習者に提供したり、あるいは、学習者が必要に応じて参照できるテキスト教材を用意すれば良いわけで、教授者にとって対応はさほど難しくない。

著者らは、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、手順に係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る(岡本ら 2013b)。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るが、こうした軽微な改善で顕著な学習効果を得ている。

##### 4.2. 「作業を介した理解」に係るつまずきと対応する学習方略

「作業を介した理解」については、大別して 2 つのつまずき要素が内在していると著者らは考える。ひとつは、具体例の認知の困難性に伴うものであり、もうひとつは、当該学習課題そのものの認知の困難性に伴うつまずきである。本章では、この類型に従って対応する学習方略を紹介する。

###### 4.2.1. 「作業を介した理解」に係るつまずきにおいて具体例の認知の困難性を伴う場合

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のとして認識されるよう明示されていることが望ましい。しかしながら、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出来ないものが散見される。この場合、学習者に対し、潜在的な動作について推察する思考過程を要求することになる。推察や判別については、それを行うか否かを含め、学習者個人の習慣や類似する先行知識の有無、そして意思などを含めた総合的能力に依存するものであるし、加えて、推察や判別それ自体が認知的負荷となって学習者の認知資源(cognitive resource)を余計に消費することになる。具体例を工夫することによってそれが直観的に理解できる対象に改善できるのであれば、推察や判別の要素は、学習者にとって非本質的な認知的負荷だったといえる。当然ながら、認知資源の多くを学習課題の本質的部分を理解するために充てるべきであって、推察や判別の要素についてはできるだけ排除することが望ましい。

こうした視点から、著者らは、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し、「視覚的顕在化」という概念(表 2 参照)にしたがって、非本質的な認知的負荷の排除を試みた教材を開発し、同教材を用いた実践でその効果を確認している(岡本ら 2013a)。

また、数学教育の分野では、Kaminski ら(2008)が、「具体的事例から抽象的要素を抽出することが難しく、事例のもつ物語性が概念の抽出を阻害している」と考察しているが、サンプルプログラムが具体例の学習で

あるという側面を捉えるならば、写経型学習にも同様の困難性が想定される。

#### 4.2.2. 「作業を介した理解」に係るつまづ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合

本項では、当該学習課題そのものの認知の困難性に伴うつまづ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する。

河内谷 (2006) は、特に 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」ことがその主たる要因ではないかと考察している。さらに、河内谷 (2006) 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る。

また、長谷川らは、プログラムの動作を条件によって切り替えたり、同じ動作を繰り返して行うなどの「制御構造」について、そのイメージを自由に絵で表現する方法で、学習者のイメージ（二次元抽象図）を調査した結果、プログラミングを習得するには、プログラムが実行された際に処理の流れの明確なイメージを持っている、つまり、実行時の動きを抽象化して捉えたイメージを持っている学習者ほど、プログラミングの理解度が高いことを示している（長谷川・山住 1997）。これに続いて長谷川らは、初学者を対象とした授業のなかで、制御構造を表す図を提示し、処理の流れのイメージを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る（長谷川・山住 1998）。

さらに、岡本ら (2013a) は、制御構造の条件分岐処理を学習する際に実物のスイッチを組み込んだマイコンボードを使用しているが、これは、実物のスイッチが条件分岐の概念を容易にイメージさせるとの考えから採用したもので、類似のイメージを提示することによって本質的な認知的負荷を軽減させようという試みである。

#### 5. おわりに

本稿では、プログラミング学習の初学者が「写経型学習過程」において見られるつまづき事例を「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」と「写経型学習の過程から目的とする内容を学び取ることができない場合」の2つに類型化し、さらに、認知的負荷理論を用いて、(1) 本質的な認知的負荷を伴う学習過程によるつまづき、(2) 非本質的な認知的負荷を伴う学習過程によるつまづきに分けて対応する学習方略を紹介および提示している。とりわけ、非本質的な認知的負荷については、表1に示したように教材の使いづらさや説明不足が要因となって生じる場合が散見されることから、手順などについては記載のとおり実行するだけでよいように提示したり、具体例を適切に選択するなどの工夫によってその排除を試みることを提案したい。

#### 参考文献

- 阿部圭一 (2009) 「C 言語によるプログラミング教育についての省察」『情報処理学会研究報告』コンピュータと教育研究会報告, 15: 205-212
- Anderson, J. R. (1985) [Cognitive psychology and its implications (2nd ed.)], New York: Freeman.
- 江木鶴子, 竹内章 (2009) 「プログラミング初心者にトレースを指導するデバッグ支援システムの開発と評価」『日本教育工学会論文誌』, 32(4): 369-381
- 深町修一 (2010) 「文系の学生に対するコンピュータプログラミング教育の一考察」『福岡国際大学紀要』, 23: 39-45
- 長谷川 聡, 山住 富也 (1997) 「プログラミング教育と学習者のイメージ形成」『名古屋文理短期大学紀要』, (22): 9-14
- 長谷川 聡, 山住 富也 (1998) 「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): 9-14
- Jennifer A. Kaminski, Vladimir M. Sloutsky, and Andrew F. Heckler (2008) [The Advantage of Abstract Examples in Learning Math]

表 2 プログラミング学習における「視覚的顕在化」(岡本ら 2013a) から引用

「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 具体的解決方法
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、一つひと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	動作が他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)	動作の頭れる場所を分離するか、他の視覚的要素から際立たせる
予測可能化	どの場所でどのような動作が実行されているかわかりにくい	3) 視認の主体が予測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か、あるいは、既存の知識や経験をもとに容易に予測できる動作の種類、動作の頭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頭れる	4) 他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一つに絞るか複数の動作をそれぞれ区別が可能なたちに分離する

『Science』, 320:454-455

河内谷幸子 (2006) 「プログラミング言語の学習法ー  
for 文の理解に関する認知心理学的分析ー」『言  
語と文化』, 法政大学言語・文化センター 編,  
3:19-35

岡本雅子, 村上正行, 喜多一, 吉川直人 (2010) 「初  
学者を対象とした自習中心のプログラミング教  
育の教材開発と評価」『情報処理学会研究報告』  
情報教育シンポジウム SSS2010 論文集: 87-94

岡本雅子, 村上正行, 吉川直人, 喜多一 (2013a) 「『視  
覚的顕在化』に着目したプログラミング学習教  
材の開発と評価」『日本教育工学会論文誌』,  
37(1): 35-45

岡本雅子, 村上正行, 吉川直人, 喜多一 (2013b) 「プ  
ログラミングの写経型学習過程を対象としたつ  
まずきの分析とテキスト教材の改善ー作業の自  
立的遂行と作業を介した理解のための支援と工  
夫ー」『京都大学高等教育研究』, 19:17-57

大島純 (2008) 「最近の認知研究からみた e- ラーニ  
ングの可能性」『The Annual Report of  
Educational Psychology in Japan』, 47:  
178-187

太田信宏 (2009) 「Java プログラミング教育に関す  
るー考察」『立教大学女子短期大学部研究紀要』  
52: 1-16

Pass, F., Tuovinen, J. E., Tabbers, H., & Vab  
Gerven, P. W. M. (2003) 「Cognitive load  
measurement as a means to advance  
cognitive load theory」『Educationl  
Psychologist』 38: 63-71

柴田望洋 (2004) 『新版 明解 C 言語入門編』, ソフト  
バンククリエイティブ.

Sweller, J., van Merrienboer, J.J.G., & Pass,  
F.(1998) 「Cognitive architecture and  
instructional design」『Educational  
Psychology Review』 10: 251-296

高橋麻奈 (2012) 『やさしい C 第 4 版』, ソフトバン  
ククリエイティブ.

## 附記

本稿は、岡本ほか (2013b) の一部について、認知  
科学的な視点を加えて再構成したものである。