

学生のプログラミングの素養を調査する手法

小林 史生*1・北 英彦*1

Email: kobayashi@ce.elec.mie-u.ac.jp

*1: 三重大学大学院 工学研究科 電気電子工学専攻

◎Key Words プログラミング教育, プログラミングの素養, 適性検査

1. はじめに

コンピュータプログラムは現代社会においてほとんど全てのシステムを動かすために使われており、その仕組み等を知ることは技術者にとって非常に大切であるため、プログラミングは多くの大学の学科で必修科目として教えられている。しかし、プログラミングは学生の生まれつきの素養によってよくできるグループとなかなか理解することができないグループに分かれることが知られている⁽¹⁾。そのような状況の中、プログラミングを教える講師はプログラミングが苦手な学生を指導することに時間を取られてしまい、その間得意な学生は授業時間を持って余してしまうということが多く起こっている。その上、プログラミングが苦手な学生は講師がどれだけ時間をかけて指導しても最後までよく理解できないというケースも多い。

このように、現状のプログラミングの授業では、プログラミングができるグループと苦手なグループが同じ授業を受けることでどちらのグループに対しても効率的な教育を実施できていないという問題が存在する。

この問題に対して、授業を実施するにあたり事前に学生のプログラミングの素養を評価することができれば、その結果と学生本人の意思を考慮して到達目標を分け、それぞれに合った教育を行うことができるようになると考えられる。

学生の素養を測る手法に関する先行研究として、S. Dehnadi らは、プログラミング学習前の学生に対して図 1 に示すような代入のプログラムを与え、意味を推測して問題を解かせ、その際に学生が一貫した考え方をしているかどうか、プログラミングの成績と強い相関関係があることを示した⁽¹⁾。しかし、この手法はプログラミングそのものを題材として扱っており、それまでにプログラミングに触れたことがある学生に対しては有効ではない可能性が高い。

そこで、本論文ではプログラミングそのものとは別の題材を用いてプログラミングの素養を評価する手法を提案する。

<p>4. Read the following statements and tick the box next to the correct answer in the next column.</p> <pre>int a = 10; int b = 20; a = b; b = a;</pre>	<p>The new values of a and b are:</p> <table style="width: 100%;"> <tr><td><input type="checkbox"/> a = 10</td><td>b = 0</td></tr> <tr><td><input type="checkbox"/> a = 10</td><td>b = 10</td></tr> <tr><td><input type="checkbox"/> a = 30</td><td>b = 50</td></tr> <tr><td><input type="checkbox"/> a = 0</td><td>b = 20</td></tr> <tr><td><input type="checkbox"/> a = 40</td><td>b = 30</td></tr> <tr><td><input type="checkbox"/> a = 30</td><td>b = 0</td></tr> <tr><td><input type="checkbox"/> a = 20</td><td>b = 20</td></tr> <tr><td><input type="checkbox"/> a = 0</td><td>b = 30</td></tr> <tr><td><input type="checkbox"/> a = 30</td><td>b = 30</td></tr> <tr><td><input type="checkbox"/> a = 10</td><td>b = 20</td></tr> <tr><td><input type="checkbox"/> a = 20</td><td>b = 10</td></tr> </table> <p>Any other values for a and b:</p> <table style="width: 100%;"> <tr><td>a =</td><td>b =</td></tr> <tr><td>a =</td><td>b =</td></tr> <tr><td>a =</td><td>b =</td></tr> </table>	<input type="checkbox"/> a = 10	b = 0	<input type="checkbox"/> a = 10	b = 10	<input type="checkbox"/> a = 30	b = 50	<input type="checkbox"/> a = 0	b = 20	<input type="checkbox"/> a = 40	b = 30	<input type="checkbox"/> a = 30	b = 0	<input type="checkbox"/> a = 20	b = 20	<input type="checkbox"/> a = 0	b = 30	<input type="checkbox"/> a = 30	b = 30	<input type="checkbox"/> a = 10	b = 20	<input type="checkbox"/> a = 20	b = 10	a =	b =	a =	b =	a =	b =
<input type="checkbox"/> a = 10	b = 0																												
<input type="checkbox"/> a = 10	b = 10																												
<input type="checkbox"/> a = 30	b = 50																												
<input type="checkbox"/> a = 0	b = 20																												
<input type="checkbox"/> a = 40	b = 30																												
<input type="checkbox"/> a = 30	b = 0																												
<input type="checkbox"/> a = 20	b = 20																												
<input type="checkbox"/> a = 0	b = 30																												
<input type="checkbox"/> a = 30	b = 30																												
<input type="checkbox"/> a = 10	b = 20																												
<input type="checkbox"/> a = 20	b = 10																												
a =	b =																												
a =	b =																												
a =	b =																												

図 1 Dehnadi らが用いた問題例

2. プログラミング学習者がつまづく点

プログラミングの授業において多くの学習者がつまづくポイントとして、以下の点が挙げられることが分かっている⁽²⁾。

- ・ 代入とシーケンス実行
- ・ 繰り返し実行
- ・ 再起実行
- ・ 並列実行

今回はこの中でも基礎的な範囲である代入とシーケンス実行、繰り返し実行に加え、同じく基礎の授業で取り扱う分岐実行を理解することができるかどうかを評価する問題を考案する。

3. Web CAB

IT 企業などが学生の採用活動等によく利用する SE の適性検査として、日本 SHL 株式会社の Web CAB というテストがある⁽⁴⁾。これは以下の 4 つのテストに加えて受験者の性格診断を行うパーソナリティ検査を実施し、SE やプログラマーといったコンピュータ職としての適性を診断するものである。

3.1 暗算

暗算のテストでは、四則演算の暗算問題を出題する。多肢選択型の問題であるが制限時間に対して問題数が多いため、数値の概算や素早い判断力が必要

となる。このテストでは、SE、プログラマだけでなくあらゆる職業に必要な計算力や素早い判断力に関係している。

3.2 法則性

法則性のテストは、図2に示すように、ある法則に基づいて図形が並んでおり、空いた部分に入る図形を選択肢から選ぶ問題である。

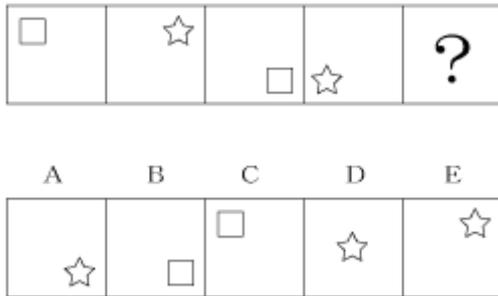


図2 法則性の問題例

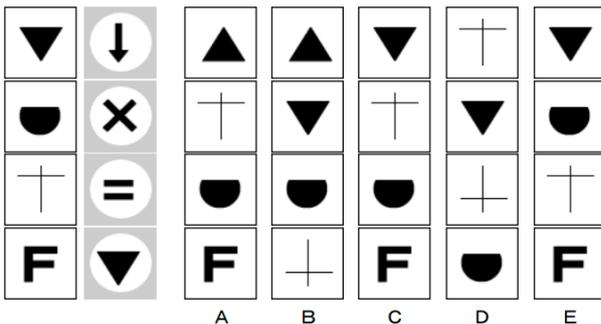
このテストは一般的にIQ (Intelligence Quotient) と呼ばれるような知能指数に関係しており、これはプログラミングを身につけて実際にプログラムを組む際に、構文等をどのように組み合わせれば目的の動作をするプログラムができるかを考える際に必要となる能力と考えられる。

3.3 命令表

命令表のテストは、図3に示すように、はじめにいくつかの命令記号が示され、与えられた図形に指示通りの命令を実行するとどうなるかを選択肢から選ぶ問題である。

	上下をさかさまにする		左右をさかさまにする
	次の命令を取り消す		前の命令を取り消す
	何も変えない		前の図形と入れかえる

(a) 命令記号の例



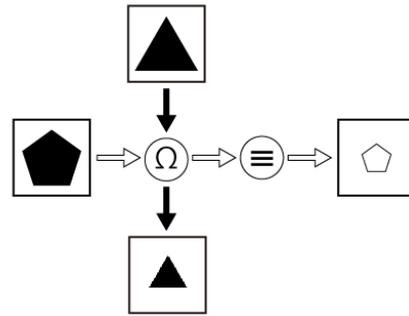
(b) 問題例

図3 命令表の問題例

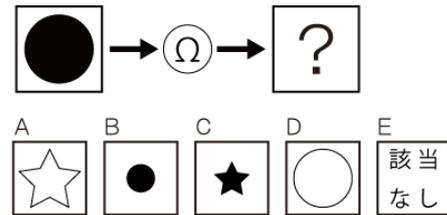
このテストは、命令記号の定義を見て、それを正しく理解して使用する能力を測ることができる。プログラミングの学習にも、初めて見る構文などの定義を理解し、それに従って記述する能力が必要であるため、このテストはプログラミングの学習段階の能力に関係の強いものといえる。

3.4 暗号

暗号のテストは、図4に示すように、暗号を適用する前後の変化を見て、その暗号の意味を把握し、後の問題の正解を選択肢から選ぶ問題である。



(a) 暗号の例



(b) 問題例

図4 暗号の問題例

このテストは、与えられた記号の変化と、その変化の要因となった暗号の関係性を把握する能力を測るもので、プログラミングにおいては他者が書いたプログラムと、それを実行した時の動作を見てプログラムのどの部分がどの動作を表しているかを把握する能力につながっている。

4. プログラミングの学習に必要な能力

プログラミングを行うのに必要とされる能力として、目的の動作を実現するためにどのような表現を用いばいいかを考えるための論理的思考能力や、発想力などがあげられるが、それらはプログラミングの基礎を学んだ後で実際に複雑なプログラムを書く際に特に必要になってくるものである。

それ以前にまずプログラミングの基礎を学習して身につけるには、初めて見るような表現の定義を正しく理解して正確に使用する能力や、もともと知っ

ていた表現に関して、それまでなじんできた意味とは異なる使い方の定義が与えられた場合にもそれを受け入れて適応していく能力が求められる。例えば、C 言語において剰余を求めるのに用いる“%”の記号はプログラミングを学び始めたばかりの人は演算子として用いるのは初めてで違和感を持つことも多いであろうし、数学において両者が等しいという意味で用いられる“=”の記号は、プログラミングにおいては代入という別の意味で使用されることが多い。このように初めて見る表現や、それまで持っていた知識と異なる定義が提示されたときに混乱してしまい順応することができない学習者は、プログラミングを学習する際に苦勞することになる可能性が高い。

このため、プログラミングの学習には、単に構文などを組み合わせて目的の機能を実現するための論理的思考能力だけでなく、決められた表現を受け入れて適応していく柔軟性も必要となってくる。

5. 素養テストの形式

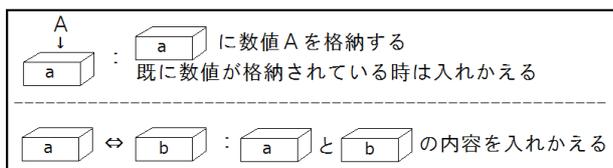
Web CAB の4種類のテストの中で、プログラミングを学習して身につける際に必要な能力に最も関係しているのは命令表のテストである。本研究では、この命令表のテストを参考にし、未知の記号に特定の命令を定義して学生に示し、それをを用いて前述の代入とシーケンス実行、分岐実行、繰り返し実行を表現した問題を作成し、学生に解いてもらうことで、その結果とプログラミングの成績の関係性を評価することを提案する。

6. 素養テストの提案

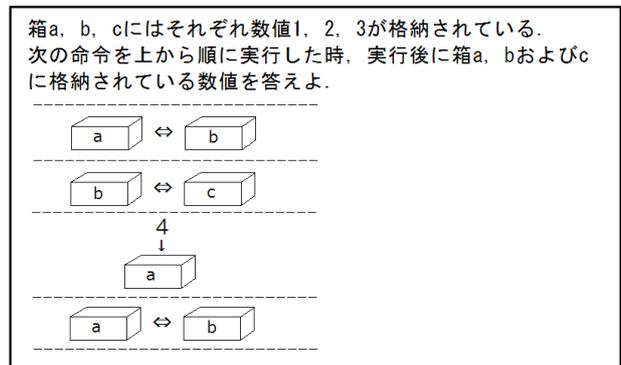
6.1 代入とシーケンス実行

代入とシーケンス実行のテストでは、図5(a)に示すように、箱に数値を格納する命令と、2つの箱に格納されている数値を入れ替える命令の表現を定義し、図5(b)に示すような問題を出題する。

この問題は、箱に数値を格納する命令が代入を、上から順番に1つずつ実行する点がシーケンス実行を表現している。



(a) 命令の定義



(b) 問題例

図5 代入とシーケンス実行のテスト

6.2 分岐実行

分岐実行を理解する能力があるかのテストとして、図6(a)に示すような命令を図6(b)の中で分岐実行として示される順序に従って処理する問題を出題する。分岐実行の問題例を図7に示す。

このテストは、分岐実行の命令の直前の演算結果が偶数か奇数かによって次の命令が分かれるという処理でプログラミングにおける if 文等に近い処理を表現している。なお、図6(b)の分岐実行の例の2つ目のように、【】内の|の左右どちらかが空欄の場合、そちらを実行する場合は何も処理しないということの意味する。これを正しく理解できればプログラミングの分岐処理も理解することができると思われる。

A ⊙ B: A, Bの和を表す 例: 3 ⊙ 5 = 8, 7 ⊙ 5 ⊙ 10 = 22
A ⊙ B: A, Bの差の絶対値を表す 例: 3 ⊙ 5 = 2, 7 ⊙ 5 ⊙ 10 = 8
A ○ B: A, Bのうちの最大の値を表す 例: 3 ○ 5 = 5, 7 ○ 5 ○ 10 = 10

(a) 命令の定義

処理順序: 基本は左から順に処理
分岐実行 []: 直前の結果が偶数の時は の左を、奇数の時は右を実行 例: 2 [⊙ 2 ⊙ 2] = 4, 2 ⊙ 3 [⊙ 1] = 3
繰り返し実行 []: 【】内の の左の演算を の右で指定された回数繰り返す 例: 3 [⊙ 2 ⊙ 3] = 3 ⊙ 2 ⊙ 2 ⊙ 2 = 9

(b) 処理順序

図6 分岐実行、繰り返し実行の命令と処理順序

上記の命令を用いると以下の演算結果がどうなるか答えよ 4 ⊙ 1 [○ 7 ⊙ 3] = (?) 2 [⊙ 2 ⊙ 1] [○ 3 ⊙ 2] = (?)
--

図7 分岐実行の問題例

6.3 繰り返し実行

繰り返し実行のテストは分岐実行のテストは、図 6 (a) の命令を、図 6 (b) の中の繰り返し実行として示される順序に従って実行する問題を出題する。繰り返し実行の例題を図 8 に示す。

上記の命令を用いると以下の演算結果がどうなるか答えよ
 $2 \odot 4 \llbracket \odot 10 \mid \odot 2 \rrbracket = (?)$
 $7 \llbracket \odot 2 \odot 4 \mid \odot 3 \rrbracket = (?)$

図 8 繰り返し実行の問題例

このテストは、 $\llbracket \quad \rrbracket$ 内の命令を指定された回数繰り返すという処理を正しく行えるかどうかを見るもので、プログラミングにおける繰り返し実行を表現している。実際のプログラミングの繰り返し表現は条件判定式によって繰り返しを制御するが、この問題では最も基本的な一定回数の繰り返し処理に近い処理となっている。条件式によって繰り返しを判定する表現に関しては、この繰り返しの命令と 5.2 で述べた分岐実行の命令を組み合わせることによってこれに近い処理を表現することが可能であるため、そうした問題も実施する。

7. 今後の予定

7.1 学生に対する素養テストの実施

今後の予定として、三重大学工学部 電気電子工学科 1 年次の「計算機基礎工学および演習」の講義の履修者を対象に、前述のプログラミングの素養テストを学生に対して実施する。この講義の履修者は大学のプログラミングの授業を受ける前の学生である。

実施内容は、代入とシーケンス実行の問題 5 題、分岐実行の問題 5 題、繰り返し実行の問題 5 題、分岐実行と繰り返し実行を組み合わせた問題 5 題程度を予定している。

また、対象の過去のプログラミング経験等も把握するため、対象の学生に対してそれらに関するアンケートを実施する。

7.2 結果とプログラミング能力の関係性の考察

素養テストとアンケートを学生に対して実施した後、プログラミングの基礎を対象の学生に指導して九九の表を作成する程度の簡単な課題を与え、その課題の成果と素養テスト、アンケートの結果の関係性を分析する。

8. まとめ

現在のプログラミングの教育では、もともとの素養によってよくできるグループとなかなか理解でき

ないグループに分かれる。これらの学生たちが同じ授業を受けると、どちらの学生に対しても効率的な教育を行うことができない。事前に学生のプログラミングの素養を知ることができれば、その結果と学生自身の希望によって授業の到達目標を分け、それぞれに適した教育を行うことができる。

本論文では、学生がプログラミングを学習する上で躓きやすいポイントに注目し、そのポイントをプログラミングとは異なる簡単な題材を用いて表現し、それを用いた問題によって学生のプログラミングの素養を評価する手法を考案した。

今後はこのテストを学生に対して実施し、実際のプログラミングの能力と比較、検証していく必要がある。

参考文献

- (1) Saeed Dehnadi, Richard Bornat : The camel has two humps (2006)
- (2) J H Benedict and B Du Boulay : Some difficulties of learning to program (1986)
- (3) 柴田望洋 : 新版 明解 C 言語 入門編 (2004)
- (4) 「日本エス・エイチ・エル」
<<http://www.shl.co.jp/>> (2014/6/2 アクセス)
- (5) 「cab・gab 試験対策集会所」
<<http://www.hatikai.info/>> (2014/6/3 アクセス)