

プログラミング教育で育成される能力の評価結果の検討

吉田典弘^{†1 †2} 篠澤和久^{†2}

概要: プログラミング教育で育成される能力の一つとして、「手順的な処理」が出来るようになることが挙げられる。そこで、大学における一般情報教育としてプログラミングの授業を行い、この授業を受講することで、「手順的な処理」に関する能力が育成されるかを評価するための事前調査を実施した。今回は評価問題として、大学入試における「情報」の参考問題を使用した、本調査実施に向けての結果検討を示す。

A Study of the Evaluation Results for Ability to be Grown in a Programming Education

NOHIRO YOSHIDA^{†1 †2} KAZUHISA SHINOZAWA^{†2}

Abstract: One of the ability to be grown in a programming teaching include that so it is "procedure processing". there. It performs a class programming as a general information education in university, by taking this class. Ability on the "procedure processing " was conducted the preliminary survey in order to assess whether is grown. As this evaluation problem, but using a reference problem of "information" in the university entrance examination, we show the results study towards this survey.

1. はじめに

プログラミング教育について、次期学習指導要領では、小学校から高等学校まで実施されることとなっている。また、大学における一般情報処理教育においてもプログラミング教育が実施されている。プログラミング教育を行うことで、小学校から高等学校までは「手順的な自動処理」を体験させること、大学においては制作経験させることができる。「手順的な処理」に関しては、情報処理学会の情報処理教育委員会が、2005年（平成17年）10月に「日本の情報教育・情報処理教育に関する提言2005」の中で公表している。この提言では、(1) 小学校・中学校・高等学校それぞれの発達段階に応じた適切な「手順的な自動処理」の体験を持たせる。(2) 高等学校の教科「情報」に選択科目を追加することで、「手順的な自動処理」に関心を持った生徒が系統的に学べる場を設ける。(3) 大学の一般情報教育において、「手順的な自動処理」についての制作体験をさせる。また各専門において、その専門に関連した情報系科目を選択可能とする(教員養成系においては必修とする)。これからの情報教育において、「手順的な自動処理」を体験や経験されることが、どの教育機関においても重要としている。

このような提言に関して、大学の一般情報教育のプログラミング教育の実践として先行研究の報告がある[1]。この報告の中では、JavaScriptを用いたプログラミング教育を

通して、論理的思考力を育成できると示されている。そこで、このことを実証すべく、筆者らは、実際にプログラミングの授業を行い、その授業の前後で問題を解かせることで、論理的思考力の評価を行い、その結果を評価した[2]。この報告では、評価問題として、国家公務員Ⅲ種および地方公務初級における採用試験の過去問題の中から、論理的思考力を問う問題を選び評価を行った。しかし、研究会において、この問題がプログラミング教育で育成される「手順」や、手順を追っていく上で必要不可欠な時間的要因を考慮した問題となっていないので、「手順的な処理」に関する能力を評価出来ていないのではないかという指摘があった。また、評価を実施した時期が、5月と7月と間隔が長かったため、もう少し短い期間で評価問題を解かせるようにした方が良いとの意見も頂いた。そこで、これらのことを踏まえ、2014年度の前期の授業において2回目の評価を行った。評価問題としては、高等学校の教科「情報」の教科書に掲載されていた、ナンバープレース(ナンバープレ)を使用した。また、評価する時期としては、6月の受講時と7月の受講時として期間を短くした。しかし、この報告[3]においても評価問題としてナンバープレはふさわしくない等の意見を頂いた。

本報告では、プログラミング教育によって育成される論理的思考力を評価するという立場から、大学における一般情報処理教育でのプログラミング教育において、「手順的な処理」に関する能力が育成されているかを評価することとした。また、2016年9月からの開始される授業において、本調査を行うための事前調査として行っており、今回得られた結果を基に評価方法を検討することを目的としている。

^{†1} 関西学院大学 教務機構 共通教育センター

Kwansei Gakuin University
n-yoshida@kwansei.ac.jp

^{†2} 東北大学大学院情報科学研究科

Graduate School of Information Science, Tohoku University

2. 評価問題と対象

2.1 評価問題

本報告では、プログラミング教育で育成される能力として、「手順的な処理」に関する能力を評価することが目的である。前回の調査では、プログラミングの授業を受講後に論理的思考力が育成されているかを検討したが、今回の評価では「手順的な処理」だけに絞って、受講後にその能力が向上しているかを評価することとした。この理由として、プログラミング教育で育成される「論理的思考力」を評価するとした場合に、「論理的思考力」が何を示しているのかを明確に定義する必要がある。しかし、このことは壮大なテーマとなってしまうため、「論理的思考力」を定義することは難しいと考えた。そこで、プログラミング教育で育成される能力、つまり正確に動作するプログラムを作成する能力として、手続きを順番に追いつながりながらプログラムを作成できる能力を評価することとした。そして、プログラミングで利用される「繰り返し」に関する理解度が調査できる問題として、慶應義塾大学湘南藤沢キャンパスが実施した2014年度の「情報」参考問題[4]の間6（付録A参照）について、若干の変更をして利用した。

2.2 評価対象について

本報告では、対象を関西学院大学の共通教育センターで開講している情報科学科目の履修学生とした。この情報科学科目を履修しているのは、いずれも非情報系学部・学科に所属している1年生から4年生である。また、この科目は選択科目であり、履修者は希望者が多数のため抽選で決まっている。授業期間は、2016年4月～7月までの春学期の授業である。プログラミング教育を実践したクラス（Aクラス）は、科目名が「コンピュータ言語（Java）」であり、Javaを用いてプログラミング初心者向けの授業を行った。履修人数は30名であったが、5月と7月の両方の評価問題に回答したのは23名であった。また、プログラミングの授業を受講していないクラス（Bクラス）は、科目名が「コンピュータ実践（ホームページ作成）」で、初心者向けの内容であり、Webサイト制作をHTMLとCSSによって行う授業であった。履修人数は45名であったが、こちらは両方の評価問題に回答したのは38名であった。

表1 調査対象のクラスについて

	履修人数	評価人数
Aクラス プログラミングの授業（Java）	30名	23名
Bクラス プログラミングの授業を受講せず	45名	38名

3. 授業内容について

今回は本調査に向けた事前調査であるので、一つの試みとして「手順的な処理」の評価問題を意識した教え方は、授業開始当初から全く行わなかった。ただ単に、Javaによるプログラミング教育を初心者向け内容で行ったのである。このことは、特別な準備などをしなくても、プログラミング教育を実施すれば「手順的な処理」に関する能力が向上するという結果が出ると考えたからである。また、授業においても「手順的な処理」を意識させるというよりは、プログラミングとして重要事項である、「順次処理」、「条件処理」、「繰り返し処理」を理解させ、これらに配列を利用することで、プログラムをより効率良く作成できるということを理解させることを一番の目的とした。以下に、この授業のシラバスを示す。なお、第2回から第9回までは教科書[5]を使用して授業を実施した。

コンピュータ言語（Java）のシラバス

- 第1回 オリエンテーション
- 第2回 順次処理1 データの出力、入力
- 第3回 順次処理2 四則計算
- 第4回 選択処理1 if文、if～else文
- 第5回 選択処理2 switch～case文
- 第6回 繰り返し処理1 for文、while文、do～While文
- 第7回 繰り返し処理2 無限ループ、二重ループ
- 第8回 配列1 1次元配列
- 第9回 配列2 2次元配列
- 第10回 アニメーション1 図形を表示する
- 第11回 アニメーション2 図形をたくさん描く
- 第12回 アニメーション3 図形を動かす
- 第13回 アニメーション4 いくつかの図形によるアニメーション
- 第14回 アニメーション5 配列を利用してプログラムを簡潔にする

4. 評価方法・結果と考察

4.1 評価方法

「手順的な処理」を評価に関しては、2016年5月から7月の授業において、前章で示したシラバスで授業を行った中で、順次処理までの授業が終わった第4回における5月の受講時と第13回における7月の受講時において評価を実施した。評価手法は以下の方法である。

1. 「手順的な処理」についての能力が育成されたかを判断する問題として、以下に示すように、5月は問題1で実施し、7月は問題2により実施した。回答時間は、2回とも10分間とした。なお、7月の実施に当たっては、5

月の問題の正解および各自の得点について一切公表はしなかった。

2. プログラミングの授業を受講したクラスと、全く受けていないクラスについて、同じ問題を出題し、その結果を比較した。
3. 5月の問題1と7月の問題2は問題を若干変更している。理由として5月の正解や点数を学生に伝えていなくても、全く同じ問題であれば確実に点数が向上すると考えたので変更を加えた。そこで、問題2の手順Bに関しては、解答すべき箇所を1か所だけ増やしている。

問題1 (5月実施)

次の手順は1から10までの合計を計算するものである。以下の①から④に当てはまるもっとも適切な語句を下の選択肢(1)～(5)から選びなさい。

(手順)

- A. 合計sumを①と置く
- B. 足す数nが1から②までのそれぞれについて次の処理を繰り返す
- C. 処理の始め
- D. ③に④を加える
- E. 処理の終わり

※①～④の選択肢

- (1) 0
- (2) 1
- (3) 10
- (4) n
- (5) sum

※1問2.5点、計10点満点

問題2 (7月実施)

次の手順は2, 4, 6, 8...100までの合計を計算するものである。以下の①から⑤に当てはまるもっとも適切な語句を下の選択肢(1)～(6)から選びなさい。

(手順)

- A. 合計sumを①と置く
- B. 足す数nが②から③までのそれぞれについて次の処理を繰り返す
- C. 処理の始め
- D. ④に⑤を加える

E. 処理の終わり

※①～⑤の選択肢

- (1) 0
- (2) 1
- (3) 2
- (4) 100
- (5) n
- (6) sum

※1問2点、計10点満点

4.2 評価結果の予測

- ・プログラミングの授業を受講したクラスは、手順的な処理の能力が育成されるので、5月の受講時と7月の受講時の点数は、7月の方が高くなっている。
- ・この授業を受けていないクラスは、手順的な処理の能力は育成されていないので、5月と7月の点数には、大きな変化は生じない。

4.3 評価結果

手順的な処理の得点結果を表2に示す。なお、両月とも10点満点である(結果に関してはt検定を実施した)。この結果より以下のことが分かる。

- (1) 表1において、プログラミングの授業を受講したAクラスの学生は、5月の得点より7月の得点が高くなっている。0.96点上昇している。
- (2) しかし、プログラミングの授業を受講していないBクラスにおいても、7月の受講時の得点の平均が高くなっている。しかも、その点数の伸びは、プログラミングの授業を受けたクラスとほぼ変わらず、0.91点上昇している。
- (3) 問題1と問題2における各選択肢に関する正答率については、図5が5月の両クラスの結果、図6が7月の両クラスの結果である。特に正答率が低いのは、両月とも選択肢①であった。
- (4) 他の選択肢については、プログラミングの授業を受講したクラスの方が正答率が高いが、図5と図6からでは、「手順的な処理」に関して、受講したクラスの方がより成果が上がっているという結果を示していない。

表2 手順的な処理の評価結果

	5月の 得点平均	7月の 得点平均	得点差
プログラミング Aクラス	4.78	5.74	0.96
受講せず Bクラス	3.62	4.53	0.91

5月 P(T<=t) 両側 0.11
7月 P(T<=t) 両側 0.05

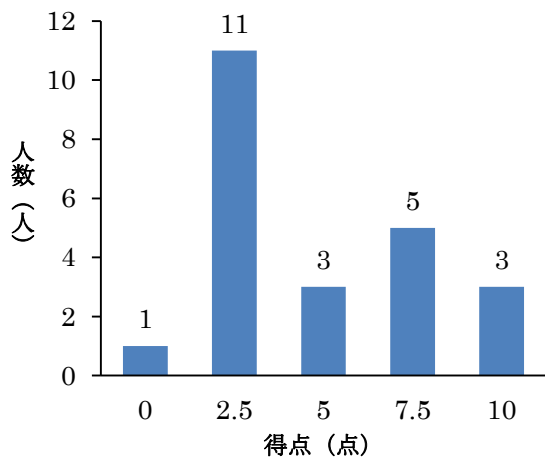


図1 プログラミング受講 (5月) 得点分布

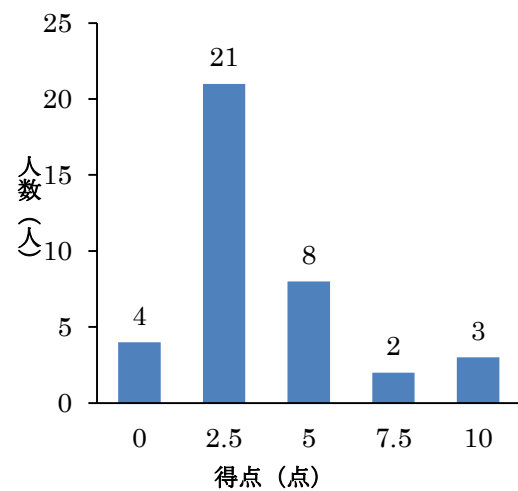


図2 受講せず (5月) 得点分布

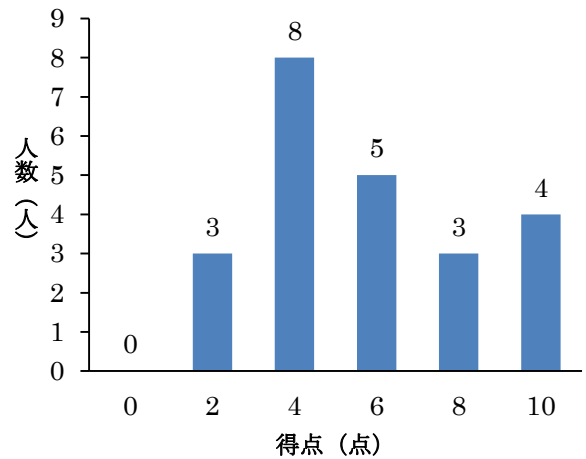


図3 プログラミング受講 (7月) 得点分布

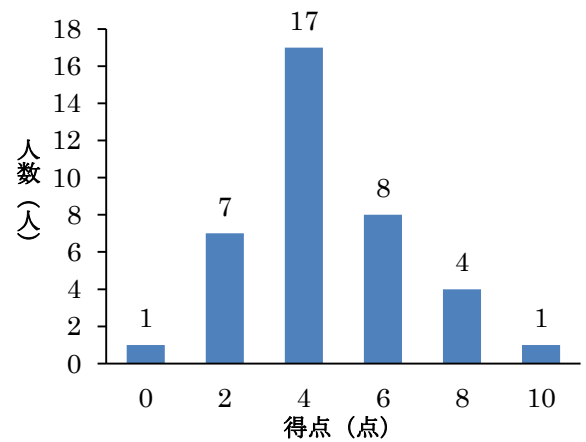


図4 受講せず (7月) 得点分布

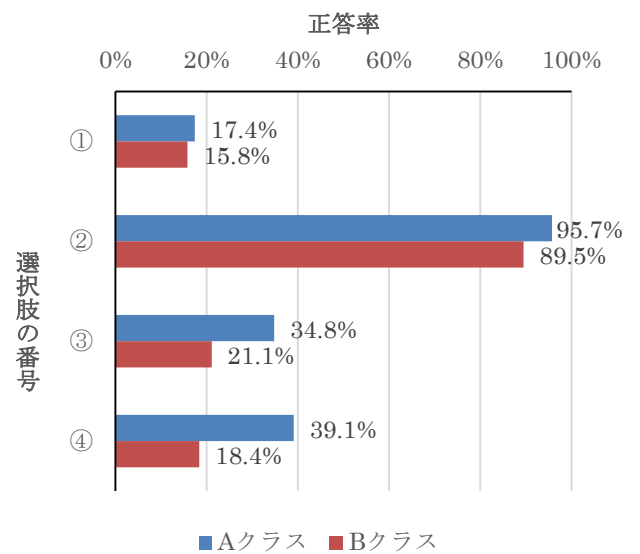


図5 問題1 (5月実施) における各選択肢の正答率

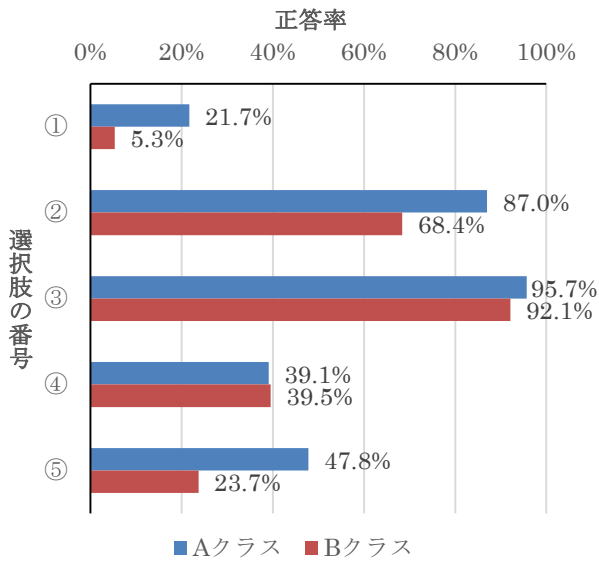


図6 問題2 (7月)における各選択肢における正答率

4.4 考察

4.4.1 評価結果に関して

図6に示す評価結果だけでは、プログラミング教育の授業を受けた学生に、「手順的な処理」に関する能力が育成されたとは、明確に言えることは出来ない。

問題点として、表1における5月の両クラスの得点差が大きいことにある。この得点差が少ない、つまり同じような得点でないと、受講したクラスの方の能力が高くなったとは言えない。本調査に向けての改善点ではあるが、現状では、この科目の登録方法や抽選方法を変更できないので、次回も1年生から4年生までの様々な学生が履修することとなる。

次に、各クラスの学生に対して、それぞれの授業を受講するまでに、プログラミング教育を受けてきたかなど、過去の学習経験に関してアンケートで確認しておく必要があった。本調査では導入する予定である。

また、各選択肢の正答率を示している図5と図6から分かることは、プログラミングの授業を受講したクラスにおいても、変数の初期化に関する設問である選択肢①での正答率が大変悪い結果となっている。また、受講したAクラスの7月の正答率が5月の正答率よりも大幅に良くなっているという結果とはならなかった。この点については次回の本調査に向けて改善する必要がある。特に変数の初期化については、点数を向上させたい。ただし、今回使用した問題において、このような記述に慣れていない場合には、文面が分かりにくかったのではないかとすることも考えられる。よって文面については変更することを検討する。

ここからは、プログラミングの授業を受けたことで、「手順的な処理」が出来るようになったことを示す結果につい

での考察を行う。まず、7月の点数が満点であった成績上位群について表2に示す。プログラミングの授業を受講したAクラスにおいて、7月の点数が10点満点であった学生が23名中4名となることが分かる。しかも、7月だけ満点が2名いた。受講していないBクラスでは、7月だけ満点はいなかったため、今回の調査で、プログラミングの授業をすることで、「手順的な処理」の能力が向上したことは示していると考えられる。

表2 10点満点の人数

	5月の満点	7月の満点	7月だけ満点
Aクラス	3	4	2
Bクラス	3	1	0

4.4.2 高校生の実施結果との比較

ここでは、慶應義塾大学の参考問題を高校生に回答させた結果と比較をする。河合塾のキミのミライ発見に掲載されている神奈川県立柏陽高等学校における実施報告である[6]。この実施での対象は「情報の科学」を履修した高校1年生で、4月から情報科の授業を始めて10月にこの試験を実施したとしている。その中から、付録Aにある参考問題を回答させた結果が図7である。なお、図7において本報告の調査結果と合わせるため、大学生の評価で行った問題における選択肢の番号を使用している。

この図7と図6を比較すると、大学生のプログラミングの授業を受講した学生の選択肢①の正答率があまりにも悪いことが分かる。これは、授業で「変数の初期化」について十分な理解が出来ていない結果となっている。授業の当初で変数の初期化については取り上げているが、今回の問題に適用できるような力が備わっていなかったと考えている。選択肢②は正答率の差は大きくないが、選択肢③と④の結果においても、図6のプログラミングの授業を受講したAクラスの正答率が下回っている。これは「繰り返し」に関する問題であるので、本調査に向けて、授業内での教授法を検討していきたい。

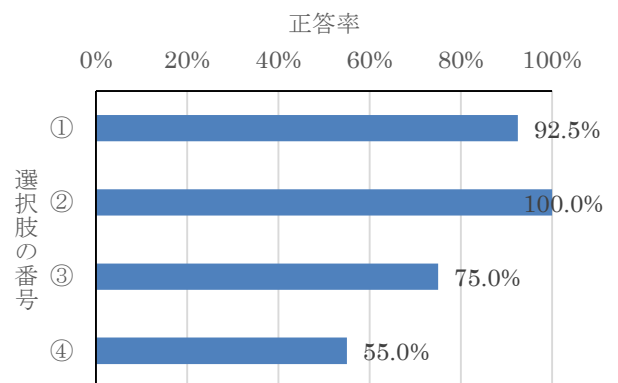


図7 高校生の各選択肢における正答率

5. おわりに

プログラミング教育を行うことで、「手順的な処理」に関する能力を評価するための事前調査を行った。「手順的な処理」に関する能力が向上したことを明確に数字として示す結果を得ることはできなかった。過去の調査と同様にプログラミング教育を行った上で、「手順的な処理」に関する能力を評価する場合、評価に相応する問題を準備し、受講後の結果として能力が向上されたことを数値として示すことは大変難しいと言える。しかし、本調査に向けて、どのような点に注意して授業を展開していけば良いかが分かった。特に授業の最初、導入時における変数の初期化については十分に配慮したい。今回は、「手順的な処理」に関する能力を評価する問題として、大学入試の情報における参考問題を使用した。本調査では問題の文面を初めて読んだ場合でも分かりやすい表現に変更をしたいと考えている。

結果の中では、プログラミングの授業を受講したクラスにおいて、成績上位群では7月の満点の学生が増加したということが判明した。ただし、これらの学生は5月の得点も満点ではないが良かった。本報告内では示さなかったが、期待したい結果としては、プログラミングの授業を受講した5月の成績下位群が7月において向上していることである。しかし、プログラミングの授業を受講していないクラスと、ほぼ同様な結果となっており、プログラミングが出来なかった学生が授業によって、手順的な処理に関する能力が向上したという結果は得られなかった。本調査に向けて、本報告で得られた結果を基に、授業の内容と評価問題について再度検討を行う。

謝辞

本研究を進める上で有益な御助言をいただいた、東北大学大学院情報科学研究科情報リテラシー教育プログラムの代表・窪俊一准教授、副代表・堀田龍也教授、静谷啓樹教授、ならびにメンバーである邑本俊亮教授、徳川直人准教授、和田裕一准教授に、謹んで感謝の意を表します。

参考文献

- 1) 河村一樹: 一般情報教育におけるプログラミング教育の在り方について、情報処理学会コンピュータと教育研究会研究報告, Vol.2011-CE108 No16, pp.1-8 (2011)
- 2) 吉田典弘, 篠澤和久: 手順的な自動処理による論理的思考力育成の評価結果の検討, 情報処理学会コンピュータと教育研究会研

究報告, CE-123-4 (2014)

- 3) 吉田典弘, 篠澤和久: 手順的な自動処理による論理的思考力育成評価結果の検討 part2, 情報処理学会コンピュータと教育研究会研究報告, CE-126-6 (2014)

- 4) 慶應義塾大学 湘南藤沢キャンパス: 一般入試「情報」参考試験 (2014年7月30日実施) の問題等の公開および実施結果について

http://www.sfc.keio.ac.jp/joho_sanko_2014_kekka.html

- 5) 照井博志: 学生ための基礎 Java, 東京電機大学出版局 (2011)

- 6) 河合塾: キミのミライ拝見,

<http://www.wakuwaku-catch.net/慶應義塾大学参考試験高校編1/>

付録 A 慶應義塾大学湘南藤沢キャンパスでの一般入試「情報」参考問題 (2014年度)

第6問

計算の手順を、文を並べて書き表すことを考える。ただし、「～の場合は次の処理を行う」、「～について次の処理を繰り返す」という文に対しては、次の処理の範囲を明確にするために「処理の始め」と「処理の終わり」という文を必ず使うものとする。「処理の始め」と「処理の終わり」は入れ子になってもよい。

(ア) 次の手順は1 から100 までの合計を計算するものである。空欄に当てはまるもっとも適切な語句を下の選択肢から選びなさい。

- A. 合計 s を17 と置く
- B. 足す数 n が1 から18 までのそれぞれについて次の処理を繰り返す
- C. 処理の始め
- D. 19 に20 を加える
- E. 処理の終わり

[17 ~ 20 の選択肢]

- (1) 0
- (2) 1
- (3) 100
- (4) n
- (5) s