

一般情報教育におけるプログラミング教育で育成される能力の分析

吉田典弘^{†1,2} 篠澤和久^{†2}

概要: 大学の一般情報教育におけるプログラミング教育で育成される能力の評価結果の分析を行った。JavaScript による動的な Web サイトの制作の授業を行い、この授業の受講前と受講後において「手順的な処理」に関する能力が育成されたかを、大学の一般入試「情報」の参考問題を利用して評価した。また、このクラスとは別にプログラミング教育を受けていないクラスでも同様な評価をした。両クラスの比較検討から、一般情報教育におけるプログラミング教育により「手順的な処理」に関する能力が育成できることを示す。

キーワード: 一般情報教育, プログラミング教育, 手順的な処理

Analysis of the ability to be trained in Programming Education for General Information Education

NORIHITO YOSHIDA^{†1,2} KAZUHISA SHINOZAWA^{†2}

Abstract: The aim of this paper is to determine whether programming education in general information education can enhance the ability of 'procedural processing'. We conducted the survey in the following way. We gave lecture on the production of dynamic website by using JavaScript in one class and then evaluated whether the students acquired their ability related to 'procedural processing' more than they had before attending the class. In the evaluation, we used the problems of 'information' in general entrance examination. We also conducted the survey using the same problems in another class, where the students did not take any programming education. From the comparison between the two classes, we show that programming education in general information education can improve the ability of 'procedural processing'.

Keywords: General Information Education, Programming Education, Procedural Processing

1. はじめに

2020 年度からの次期学習指導要領では、小学校から高等学校までプログラミング教育が実施されることとなっている。高等学校における教科「情報」での新設科目においてプログラミングを必修とされている。現在までに各教育機関でのプログラミング教育の在り方について、様々な方法で実施されているが、今後は、それぞれの教育段階でどのように評価するかが問われると考えられる。大学の一般情報教育においても、教養教育としてプログラミング教育が実施されている。大学生の低次年を対象としたプログラミング教育に関する研究では、授業の最終成績が向上するなど、その評価において、「成績」が向上したとする成果の報告はあるが、プログラミング教育の授業により、何らかの能力が育成されたことを、直接的かつ定量的に評価することは難しいとされている。一方、以前よりプログラミング教育では、小学校から大学まで「手順的な自動処理」を体験、あるいは、これを利用して教育することが望ましいとされている。「手順的な自動処理」に関しては、情報処理学会の情報処理教育委員会が、2005 (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」[1]の中で公表している。この提言では、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」を体験や経験させることが、どの教育機関においても重要としている。さらに、「手順的な自動処理」により、自分が記述したものの論理的帰結が明確に示され、機械で実行されたものとの両者の差異を確認することができるという利点が挙げられている。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の報告がある[2]。この中では、プログラミング教育と論理的思考力育成の関係が示されており、その授業実践の一例として JavaScript を用いている。そこで、プログラミング教育を受けた学生に論理的思考力が育成されることを評価したいと考え、このことを実証すべく、筆者らは、2013 年度後期に、プログラミングの授業を行い、その授業の前後で論理的思考力を評価する問題を解かせた。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、論理や真偽などを問う問題を選び評価を行った[3]。しかし、この問題がプログラミング教育で育成される「手順」や、特に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、「手順的な自動処理」に関する

^{†1} 関西学院大学 教務機構 共通教育センター
Kwansei Gakuin University

^{†2} 東北大学大学院情報科学研究科
Graduate School of Information Sciences, Tohoku University

能力の評価が出来ていなかった。また、事前事後テストによる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、もう少し短い期間の方が得点上昇などの成果が得られる可能性があった。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、4行4列のナンバープレース（ナンプレ）を使用し、その解答方法の過程を問題とした。また、事前事後テストによる評価を実施する時期としては、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[4]においても評価問題としてナンプレはプログラミング教育で育成される能力を評価することは出来なかった。

そこで、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、大学の一般情報教育における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。この考えを基に実施した2016年度前期の事前調査では、プログラミングを行った授業、行っていない授業、両方とも授業後の点数が上昇していた。この結果を踏まえ、2016年度後期に本調査を実施した。本論文では、本調査の結果を分析することで、授業回数、15回（90分間×15回）程度でも「手順的な処理」に関する能力が育成されていることを、特に事前テストの得点が下位群の履修生に注目し、事後テストでの得点で向上していることを分析することで示す。

2. 事前調査

本調査の方法および結果を検討する前に、この章では、事前調査について示す。プログラミング教育で育成される能力の評価方法が適切であるかを確認するために2016年の4月から7月の授業で実施した[5]。

2.1 評価問題

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」を評価することとした。「手順的な処理」とは、「手続きの順番を的確に記述する」ことを指し、この能力を評価するために、簡単なプログラムを作成する問題を利用することとした。プログラミングで利用される「繰り返し」に関する理解度を評価できる問題として、慶應義塾大学湘南藤沢キャンパスが実施した、2014年度の一般入試「情報」の参考問題[6]の間6（付録A）を用いることとし、プログラミングの初心者に対して、分かりやすい文面とするため、若干の変更をして利用した（付録B）。

2.2 評価対象と授業内容

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月から7月までの春学期の授業であった。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。授業はEclipseを用いたがテキスト入力をする形式で進めた。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により行う授業であった。よって、プログラミングに関する内容は一切扱っていない。履修人数および評価人数は表1示す。Javaの授業のシラバスについても掲載しておく。

表1 事前調査対象のクラスについて

	履修人数	評価人数
Aクラス (Java)	30名	23名
Bクラス (HTML&CSS)	45名	38名

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 繰返し処理1 for文、while文、do～While文
- 第7回 繰返し処理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

2.3 評価方法

「手順的な処理」の評価に関しては、2016年4月から7月の授業において、前節で示したシラバスで授業を行い、事前テストとして順次処理までの授業が終わった第4回の冒頭における5月の受講時と、事後テストとして第13回の冒頭における7月の受講時において評価を実施した。評価方法は以下の通りである。

- 5月は問題1（付録B）で実施し、7月は問題2（付録B）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。

- 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。
- 5月の問題1と7月の問題2は問題を若干変更し難易度を高くしている。この理由として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からである。よって、問題2の手順Bに関して、解答すべき箇所を1か所だけ増やしている。

2.4 評価結果

表2に、それぞれのクラスの5月と7月における平均点と得点差を示す。この事前調査では、プログラミングの授業を受けたクラスの得点は向上しているが、受講していないクラスも同様に向上していった。また、表3は手順的な処理の評価結果として、平均と標準偏差をまとめたものである。表4は事前事後テストに関する2要因分散分析の結果である。分散分析にはフリーソフトのjs-STARを用いた[7]。

表2 手順的な処理の得点結果

	5月の 得点平均	7月の 得点平均	得点差
Aクラス (Java)	4.78	5.74	0.96
Bクラス (HTML&CSS)	3.62	4.53	0.91

表3 事前調査の結果

	Java 群		HTML&CSS 群	
	事前	事後	事前	事後
N	23	23	38	38
Mean	4.78	5.74	3.62	4.53
S.D.	2.94	2.59	2.55	2.09

表4 事前調査における分散分析の結果

S. V	SS	df	MS	F
A	40.4772	1	40.4772	4.52 *
subj	528.8465	59	8.9635	
B	24.9022	1	24.9022	6.22 *
AxB	0.0169	1	0.0169	0.00 ns
sxB	236.1921	59	4.0033	
Total	830.4350	121		+p<.10 *p<.05 **p<.01

表4より、授業前後の主効果のみが有意であった ($F(1, 61) = 6.22, P < 0.05$)。表3より事前、事後テストの平均を比べると事後テストの方が大きく、両クラスとも手順的な処理についての理解度を促進している効果を示したといえる。しかし、授業間の効果の差は見出されなかった。

3. 本調査

本調査は、事前調査で両クラスの分散分析の主効果が有意であったので、2016年の9月から12月の授業において、同様な方法で実施した。本節では、この調査方法と結果を示す。

3.1 評価対象

対象は、事前調査と同様に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。なお、今回のプログラミングのクラスは、コンピュータ実践（ホームページ作成）、プログラミングを受講していないクラスはコンピュータ実践（表計算）である。事前調査とは授業名が異なるが、プログラミングの授業とそれ以外の授業の比較するために、この授業を選択した。表4に各クラスの履修人数と評価人数を示す。

表4 本調査対象について

	履修人数	評価人数
Cクラス (JavaScript)	30名	25名
Dクラス (表計算)	45名	36名

3.2 授業内容

JavaScriptの授業は、エディタとしてTerapad、それをブラウザで確認するというテキスト入力を行う方法で実施した。授業の目的は、動的なWebサイトの作成の基礎を身に付けることとし、プログラミング教育を初心者向けの内容として実施した。また、テキスト入力としているのは、このような環境であっても、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結果が出るという考えに基づいている。授業においては「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基本事項である、「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。また、授業で使用した例題を掲載した(付録C)。なお、教科書[8]を使用して授業を実施した。表計算クラスは、表計算ソフトのExcelの応用学ぶクラスであり、列の検索や、条件付き集計を関数で求められる、ピボットテーブルを利用して集計できることを最終目標とした。

コンピュータ実践 (HP 作成 : JavaScript) のシラバス

- 第1回 オリエンテーション
- 第2回 HTML の基礎 画像の表示, ハイパーリンクの設定
- 第3回 イベントハンドラの利用, 関数の基本, 変数の利用, 変数の演算
- 第4回 配列, 関数の引数, 繰り返し処理, 文字の表示
- 第5回 条件分岐 if文, if~else文, switch文
- 第6回 breakとContinue, 関数の戻り値
- 第7回 文字入力とエラー処理, オブジェクト
- 第8回 ウィンドウの操作, スクロールの操作
- 第9回 文字色, 背景色の操作, 画像の操作
- 第10回 日付, 時間の操作, Math オブジェクト
- 第11回 一定間隔での処理を繰り返す, 簡単なアニメーション
- 第12回 最終課題1 各自で15枚以上の静止画を利用してアニメーションを作成
- 第13回 最終課題2 各自で15枚以上の静止画を利用してアニメーションを作成 (第12回の継続)

3.3 評価方法

評価問題の実施は, 2016年9月から12月の授業において, 前節で示したシラバスで授業を実施した. 事前テストとして第4回の冒頭における10月の受講時と, 事後テストとして第12回の冒頭における12月の受講時において評価を実施した. 評価手法は事前調査と同じである.

3.4 事前調査からの改善点

事前調査では, 変数の初期化に関する得点が非常に悪かった. そこで, 本調査のJavaScriptの授業内では, 変数の初期化と, 繰り返しに関する理解度を高めるように, 授業内で繰り返しの部分を中心に詳細な説明を行った. ただし, 評価問題に関する内容は扱っていない.

3.5 結果の予測

- JavaScriptによる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, 10月の受講時と12月の受講時の点数は, 12月の方が高くなっている.
- この授業を受けていない, 表計算クラスは, 「手順的な処理」の能力は育成されていないので, 10月と12月の点数には, 大きな変化は生じない.

3.6 評価結果

「手順的な処理」の得点結果について, 両クラスの得点分布を図1から図4, 各問題番号の正答率を図5と図6, 統計解析の結果を表5から表7に示す. 表6は, 手順的な処理における平均と標準偏差をまとめたものである. 表7は事前事後テストの評価としての2要因分散分析の結果である. 以下, 結果から得られた事項である.

- 両クラスの得点分布を比較すると, それぞれ授業後の図2と図4において, 得点が向上している履修生が多いことが分かる.
- 表5において, プログラミングの授業を受講したCクラスの履修生の平均点は, 10月の得点より12月の得点が高くなっており, 1.04点上昇している. 3) プログラミングの授業を受講していないDクラスにおいては, 12月の得点の平均は高くなっているが0.17点であり, 大幅な上昇ではない.
- 問題1と問題2における各問題番号に関する正答率で, 特に正答率が低いのは, 両月とも問題番号①である.
- 他の問題番号については, 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はいない.
- 表7より, 授業前後の主効果に有意はなかった. 表6よりプログラミングの授業における事前, 事後テストの平均を比べると事後テストの方が大きく, 手順的な処理についての理解度効果を示したといえるが, 授業間の効果の差は見出されなかった.

表5 本調査における手順的な処理の評価結果

	10月の 得点平均	12月の得 点平均	得点差
Cクラス (JavaScript)	4.00	5.04	1.04
Dクラス (表計算)	3.89	4.06	0.17

表6 本調査の結果

	JavaScript群		表計算群	
	事前	事後	事前	事後
N	25	25	36	36
Mean	4.00	5.04	3.89	4.06
S.D.	2.55	2.66	2.66	2.08

表7 本調査における分散分析の結果

S.V	SS	df	MS	F
C	8.8542	1	8.8542	1.15 ns
subj	453.6744	59	7.6894	
D	10.7413	1	10.7413	2.13 ns
CxD	5.6266	1	5.6266	1.12 ns
sxD	297.2300	59	5.0378	
Total	761.0532	121		+p<.10 *p<.05 **p<.01

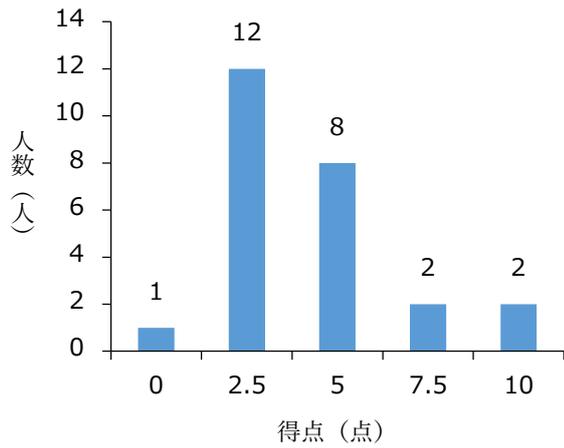


図1 JavaScript クラス得点分布 (10月)

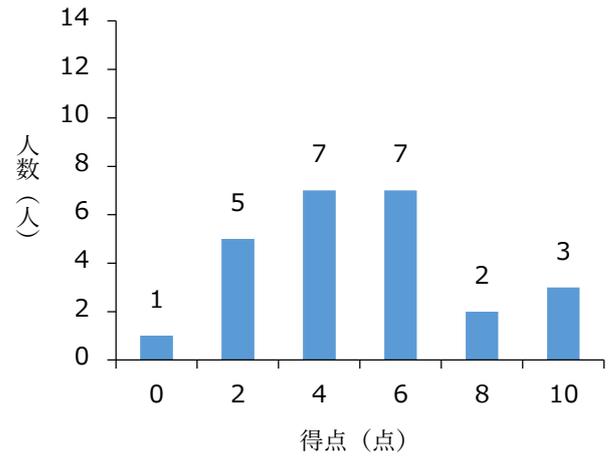


図2 JavaScript クラス得点分布 (12月)

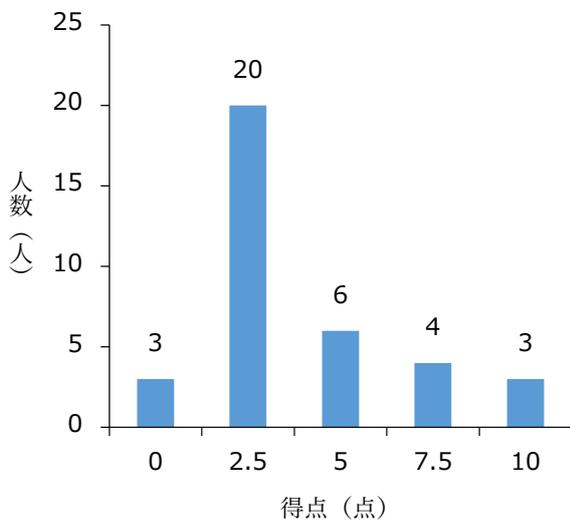


図3 表計算クラス得点分布 (10月)

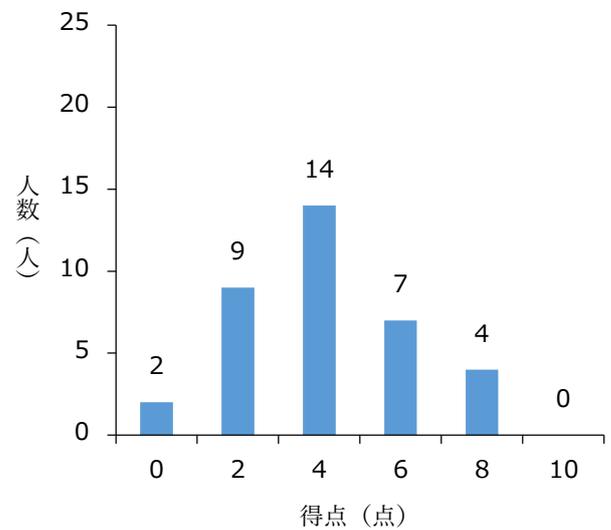


図4 表計算クラス得点分布 (12月)

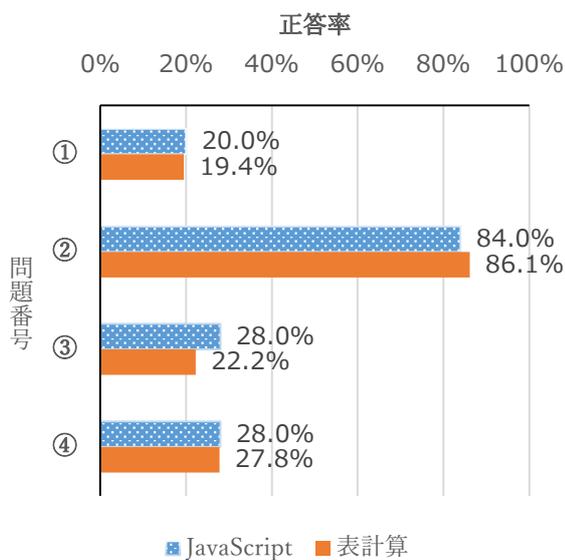


図5 10月における各問題番号の正答率

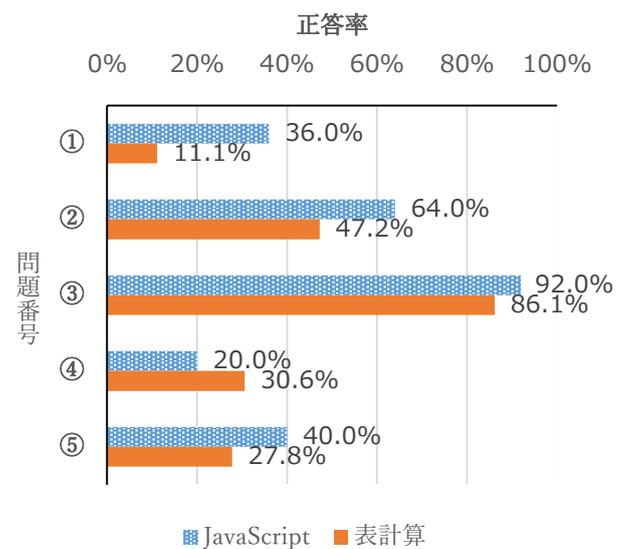


図6 12月における各問題番号の正答率

4. 考察

4.1 全体の結果

表5のように、プログラミング教育の授業を受けたクラスの得点は大きく向上し、受けていないクラスの得点との差が出た。しかし、表7の分散解析では有意な結果であることを示すことが出来なかった。よって、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は言えないが、事前調査と同様に、本調査でも事後テストの平均点が向上することは示せた。

また、各問題番号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問題番号①での正答率が大変低い結果となっていることである。プログラミングのクラスの問題番号①の12月の正答率が10月の正答率よりも高くはなっているが、大幅に良くなっているという結果とはならなかった。

次に、表5に示したように、プログラミングのクラスの得点の平均点は向上しているが、図2の得点分布のように、全履修生の得点が向上しているわけではなく、10月が高得点であっても、12月に得点が下がってしまう履修生もいた。この点は、事前調査でも同じであったが、事後テスト実施時における、履修生のモチベーションを維持することが難しいことを示している。

「手順的な処理」の能力を評価する上では、問題1であれば問題③と④、問題2であれば問題番号④と⑤の正答率について見るべきである。図6において、問題番号④の正答率が表計算クラスの方が高くなっているが、問題番号⑤では、プログラミングのクラスの正答率が高い。しかし、両方正答率を合わせても、プログラミングによる能力の育成が向上したといえるほど、正答率は高くなかった。

4.2 得点群による評価

次に、プログラミングの授業を受けたことで、受講開始時に「手順的な処理」が出来なかった学生が、出来るようになったことを示す結果として、得点群による評価を行った。得点群として、10月の得点が、「0点と2.5点」、「5点」、「7.5点と10点」の3群に分け、それぞれの平均点が12月にどのように変化したのかを示したのが、図7（JavaScriptクラス）および図8（表計算クラス）である。図7より、10月の得点が下位群（0点と2.5点）において、平均点の向上がみられる。また表8と表9より、JavaScriptの得点下位群は、14名中11名の得点が向上し、平均点が2.14点から4.86点となった。表計算のクラスは、23名中17名の得点が向上し、23名の得点の平均点が、2.17点から3.91点となった。表9は10月の得点下位群だけの統計解析、表10には、得点下位群だけの事前事後テストに対する2要因分散分析の結果を示す。

表8 10月得点下位群の得点上昇

	10月の得点下位群 (人)	得点上昇者 (人)	得点上昇 (点)
JavaScript	14	11	2.72
表計算	23	17	1.74

表9 10月得点の下位群における分析の結果

	JavaScript 群		表計算群	
	事前	事後	事前	事後
N	14	14	23	23
Mean	2.14	4.86	2.17	3.91
S.D.	0.87	2.80	0.84	2.17

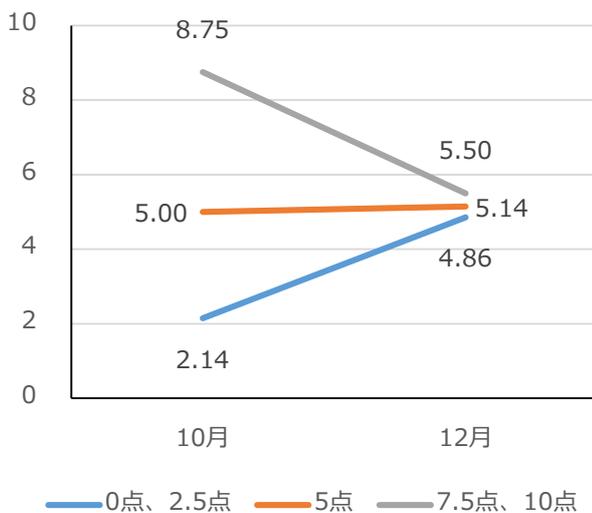


図7 得点群における平均点 (JavaScript)

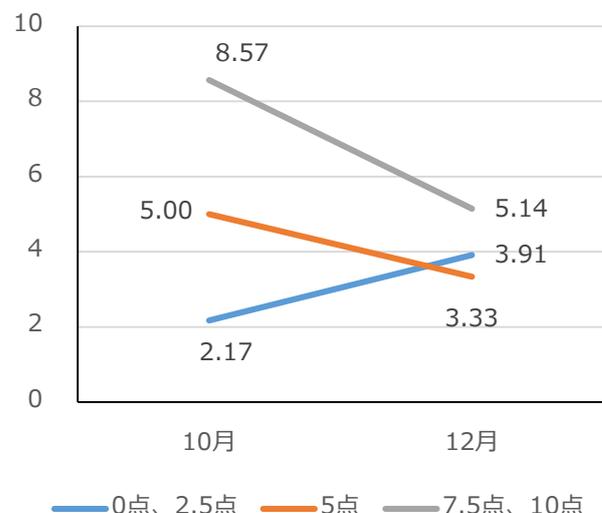


図8 得点群における平均点 (表計算)

表 10 10月の得点の下位群における分散分析

S. V	SS	df	MS	F
CL	3.6275	1	3.6275	0.97 ns
subj	130.9130	35	3.7404	
DL	86.3000	1	86.3000	26.58 **
CLxDL	4.1378	1	4.1378	1.27 ns
sxDL	113.6460	35	3.2470	
Total	338.6243	73		+p<.10 *p<.05 **p<.01

※CLはJavaScript得点下位群, DLは表計算得点下位群

表 10 より, 授業前後の主効果のみが有意であった。(F(1, 37)=26.58, P<0.01). 事前, 事後テストの平均を比べると事後テストの方が大きく, 両クラスとも手順的な処理についての理解度を促進している効果を示したといえる. しかし, 授業間の効果の差は見出されなかった.

また, 図 7 において, JavaScript クラスの 10 月の上位群および中位群の平均点が 12 月に下落しているのは, 図 8 の表計算にも同様なことが言える. これは, 12 月の実施時に履修生の問題への取り組み方を確認していたが, 授業の成績には関係はないとしたため, 前節でも述べたように問題への取り組みに関してモチベーションが大幅に下がっていることにあると考えられる.

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本調査を実施した. 「手順的な処理」に関する能力が向上したとする, 授業前後の平均点の向上および事前テストにおける成績下位群の平均点の向上は示せた. しかし, 交互作用は有意ではなかった. 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に有意な結果として得ることは大変難しいと言える. しかし, 事前調査と本調査の分析結果により, プログラミング教育を行うことで育成される能力を直接的に評価する上で, 「手順的な処理」に関する評価が有効である可能性を示した.

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]情報処理学会情報処理教育委員会: “日本の情報教育・情報処理教育に関する提言 2005“, <https://www.ipsj.or.jp/12kyoiku/proposal-20051029.html> (2017年2月15日確認)
- [2]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)
- [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, pp.1-6 (2014)
- [4]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, pp.1-8 (2014)
- [5]吉田典弘, 篠澤和久: “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“, 情報処理学会コンピュータと教育研究会研究報告, CE-136-9, pp.1-6 (2016)
- [6]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 (2016年4月23日確認)
- [7]中野博幸, 田中敏: “フリーソフト js-STAR でかんたん統計データ分析“, 技術評論社 (2012)
- [8]相澤裕介: “(新) JavaScript ワークブック“, カットシステム (2011)

付録

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.

- 合計 s を 17 と置く
- 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
- 処理の始め
- 19 に 20 を加える
- 処理の終わり

[17 ~ 20 の選択肢]

- (1) 0
- (2) 1
- (3) 100

- (4) n
- (5) s

付録 B 「手順的な処理」の評価問題

問題 1 (事前テスト)

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. 合計を入れる変数 sum を①と置く
- B. 足す数の変数を n として、この数字が 1 から②まで、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

※各 2.5 点, 計 10 点満点

問題 2 (事後テスト)

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. 合計を入れる変数 sum を①と置く
- B. 足す数の変数を n として、この数字が②から③まで、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

※各 2 点, 計 10 点満点

付録 C JavaScript の授業での例題(九九の表を作成する)

```
<HTML>

<HEAD>
<TITLE>繰り返し処理</TITLE>
<META http-equiv="Content-Type" content="text/html;
charset=Shift_JIS">
<STYLE type="text/css">
<!--
TH{background-color:#CCCCCC;}
TD,TH{
width:50px;
text-align:center;
}
-->
</STYLE>
</HEAD>

<BODY>
<H3>九九の表</H3>
<TABLE border="2">
<TR>
<TH></TH><TH>1</TH><TH>2</TH><TH>3</TH><TH>4</TH><TH>5</TH><TH>6</TH><TH>7</TH><TH>8</TH><TH>9</TH>
</TR>
<SCRIPT type="text/javascript">
<!--
for (i=1 ; i<=9 ; i++){
document.write("<TR><TH>" + i + "</TH>");
for (j=1 ; j<=9 ; j++){
document.write("<TD>" + i*j + "</TD>");
}
document.write("</TR>");
}
//-->
</SCRIPT>
</TABLE>
</BODY>

</HTML>
```