

Programming Education as General Information Education in Higher Education through practices in Hokkaido University

Izumi Fuse^{1,2,4)*} and Shigeto Okabe^{3,4)}

1) Information Initiative Center, Hokkaido University

2) Center for Open Education, Hokkaido University

3) Professor Emeritus, Hokkaido University

4) Institute for the Advancement for Higher Education, Hokkaido University

高等教育の一般情報教育におけるプログラミング教育 —北海道大学の実践を通して—

布施 泉^{1,2,4)**}, 岡部 成玄^{3,4)}

1) 北海道大学情報基盤センター

2) 北海道大学オープンエデュケーションセンター

3) 北海道大学名誉教授

4) 北海道大学高等教育推進機構研究員

Abstract — In the development of information technology, programming education has been promoted worldwide in elementary and secondary education. For instance, the UK started education in the compulsory subject “computing” in elementary and secondary schools from September 2014. The purpose of this paper is to analyze the results of information education, especially programming education, as information education in general education of Hokkaido University over the last ten years and more and to discuss the purpose of programming education, possible forms of its content and programming languages used in connection with elementary and secondary school information education.

(Accepted on 8 February, 2016)

*) Correspondence: Information Initiative Center, Hokkaido University, Sapporo 060-0811, Japan

**) 連絡先：060-0811 札幌市北区北 11 条西 5 丁目 北海道大学情報基盤センター

1. はじめに

近年の情報技術の進展の中、初等中等教育段階におけるコンピューティング教育の充実が世界的に推進されている。例えば英国では、2014年9月から、小学校段階からのコンピューティング教育を必修で実施しており、コンピュータ科学の原理・概念の理解や問題解決のためにプログラミングを使う実践的な経験を積み重ねる学習を行っている¹⁾。我が国でも、プログラミング教育の充実が国として言及されており²⁾、今後、ますます推進されていくことが予想される。

本論文では、初等中等教育段階の状況を踏まえ、高等教育の一般教育としての情報教育の学習構成におけるプログラミング教育の位置づけを、目的、レベル、および言語等から検討することを目的とする。

2015年現在の学習指導要領では、中学校・技術分野における「情報に関する技術」として、「プログラムによる計測と制御」が必修化されており、「コンピュータを利用した計測・制御の基本的な仕組みを知ること」「情報処理の手順を考え、簡単なプログラムが作成できること」が全生徒に指導されることとなっている。一方、高等学校の共通教科「情報」は、「情報の科学」「社会と情報」の2科目から構成される。「情報の科学」では、「問題解決とコンピュータの活用」にて、処理手順の自動化やモデル化とシミュレーション等が取り上げられ、「学校や生徒の実態に応じて適切なアプリケーションソフトウェアやプログラム言語を選択すること」と学習指導要領に記載されており、プログラミングを取り上げるはずである。しかし、「社会と情報」では、プログラミングについてあらわには記載されてはおらず、高々問題解決として、情報の収集・整理や評価をさせる際に情報機器を用いることの有用性を理解させることが言及されている程度である。これら「情報の科学」や「社会と情報」を履修した学生は、2016年度から大学に入学する。また、2018年度からは、中学校・高等学校ともに現行学習指導要領の履修者が大学に入学する。従ってその時期以降では、原則的には、プログラミングを全学生が経験した上で大学に入学するものと期待される。

2015年度現在、大学に在籍する学部学生は概ね、

前学習指導要領である中学校技術分野「情報とコンピュータ」、および普通教科「情報」を履修している。前学習指導要領では、中学校における「プログラムと計測・制御」は選択履修であり、その履修割合が問題となる。また、高等学校「情報」は、「情報A」「情報B」「情報C」科目のどれかを履修して入学する者が殆どである。後述するように、本学における入学時の全数調査では、中学校および高等学校でのプログラミングの実施割合は、高々15%–18%程度と低いことが示されている。

大学の一般教育としての情報教育については、情報処理学会一般情報教育委員会が、2013年から2014年にかけて、実態調査を行っており、結果は、情報処理学会誌で報告されている(岡部 2014)。全国の対象大学の3割強の回答率の結果である。ほぼ全学生が履修する必修もしくは必修相当の一般情報教育科目が9割強の大学で設置され、さらに選択科目は、半数の大学が設置しているとある。これら回答科目全体の1割が科目全体としてプログラミングを取り上げており、1回から数回と部分的にプログラミングを行っている科目を合わせると3割程度との結果である。このように、大学における一般情報教育として、一部でもプログラミングを取り上げている大学は少数派であることから、現状では、プログラミングに全く触れないまま、大学を卒業する学生が一定割合以上存在することが懸念される。

北海道大学(以下、本学)では、2001年度から2005年度は、一般情報教育科目が3科目で構成されていた。うち2科目(必修相当1科目、選択1科目)は実習形態で行い、必修相当の科目では数回のプログラミング教育を、選択科目では科目全体でプログラミング教育を行ってきた。前者はFortran, JavaScript, Cから担当者により1言語を選択し、後者ではRubyを取りあげた。2006年度からは、単位の実質化に伴う科目数減の要請により、また、高等学校で普通教科「情報」履修者が入学することを踏まえ、科目内容も前期実習型(必修)の「情報学Ⅰ」と後期座学型(選択)の「情報学Ⅱ」の各2単位2科目構成に改訂した。プログラミング教育が不要になった訳ではないものの、前期の実習型授業では、高等学校での学習を踏まえ、情報活用能力のさらなる深化育成を目指し、情報学の基礎的内容の他、情

報倫理教育、情報セキュリティ教育ならびにグループでの協調学習を組み合わせたカリキュラムを実施しており、プログラミング教育を単体として割ける時間は殆どないのが実状である。このため、科目全体として行う Ruby のプログラミング教育は、一般選択科目に移行することとし、現在まで継続して実施している（本学は1学年が2600名程度である。他の一般科目と競合する中での選択履修であり、かつ1単位であるにも関わらず、履修者は毎年100名前後である）。一方、必修の情報学Ⅰでは、2009年度から、授業時1コマの一部で、プログラミングをあらわに取り上げ、現在は、課題と小テストを課すことでプログラミング体験とその理解度を評価することとしている。更に、他の学習項目の中でも、プログラミングをツールとして用いて問題解決にあたる課題を付与している。選択の情報学Ⅱでは、プログラミング教育を一部取り上げている。本論文では、このような一般情報教育での授業構成におけるプログラミング教育の要素の組み込みについても検討することとする。

以下、第2章では本学の必修の一般情報教育におけるプログラミングの実践内容を学習者の状況を含めて報告する。第3章では実践結果からの分析と検討を行う。第4章での考察を踏まえ、今後の一般情報教育におけるプログラミング教育の方向性を第5章でまとめる。

2. 北海道大学における一般情報教育でのプログラミング教育の実践

2.1. 入学時調査から見える学生を取り巻く状況の変化

高等学校で、新教科として、普通教科「情報」が実施されたのが2003年度である。当初、町のパソコン教室以下といった揶揄もなされていたが、2006年度に本学に入学した学生への入学時調査によると、コンピュータ操作に関わるスキルの向上は顕著に見受けられた。本調査は、必修の情報教育科目の第1回授業時に、毎年、学生に行っているものである。図1に2004年度から2015年度までの年次変化を示す。2006年度は新旧課程の対象者を分け、(新)

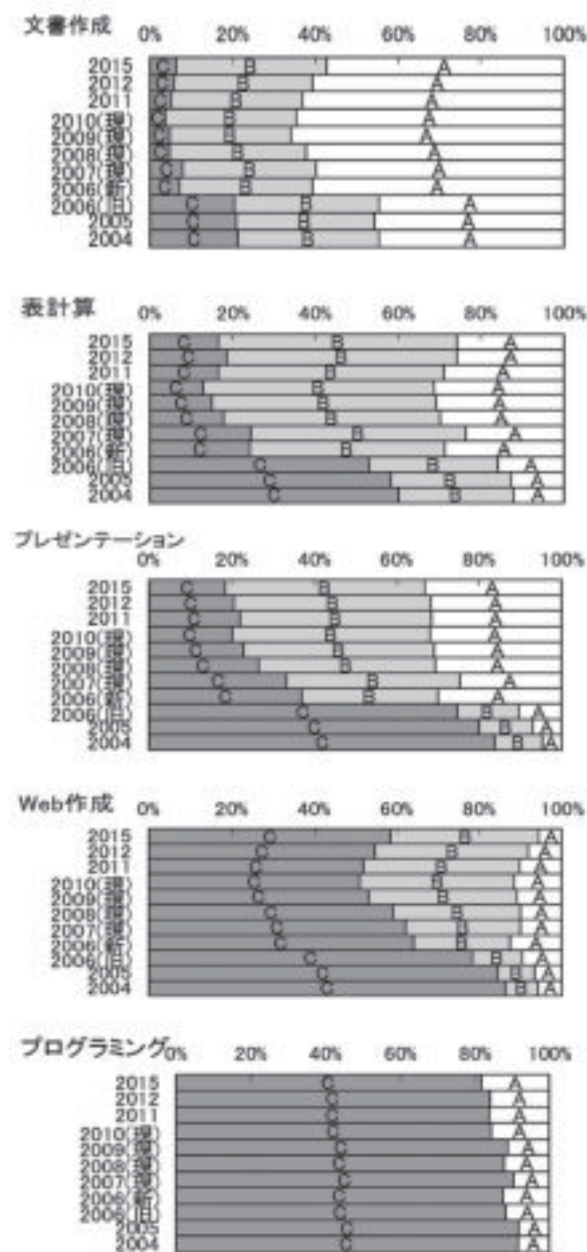


図1. リテラシ能力についての年次変化

(旧)と表示した。2007年度から2010年度は各年の現役学生のみを抽出して表示した（現と表記）。2010年度において、現役・浪人の別で結果が依存しないことを確認した上で、2011年度以降は全入学生をひとまとめで示した。なお、2013、2014年度は同傾向のため割愛した。本学では2600名程の入学生がおり、その96-98%を網羅した結果である。文書作成、表計算、プレゼンテーション、Webページ作成では、「できる」「大体できる」「自信がない」「できない」の4択で自己評価させた結果を、「できる」「大体できる」をA、「自信がない」をB、「できない」をCとしてまとめた。2006年度を境に、新課程の

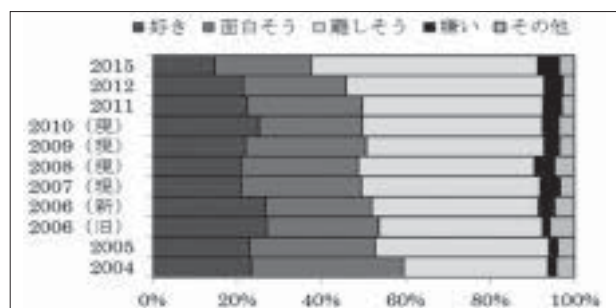


図2. コンピュータの印象についての年次変化

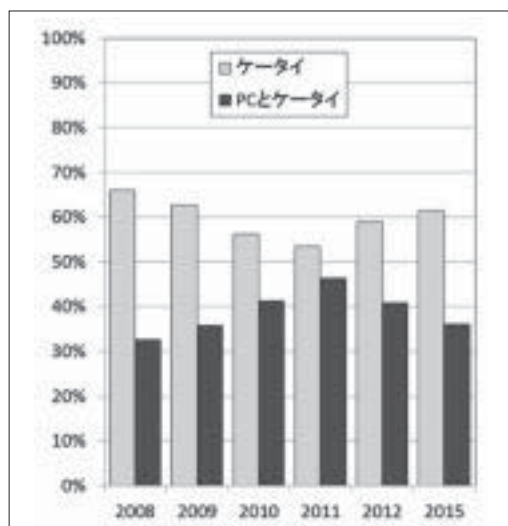


図3. メール利用の際の情報端末の年次変化（ケータイはスマートフォンを含む携帯型情報通信端末）

履修者において、「できない」割合の減少、および「できる・大体できる」割合の増大が見て取れる。また、2008年度から、学生は中学校・技術分野でも「情報とコンピュータ」を履修しており、2006年度に比べると小さな変化であるが、「できない」割合がさらに減少している。

図1で、プログラミングに関しては、経験の有無を聞いている。「できる、している・経験がある」をA、「経験がない」をCとして図示した。この傾向は、特に2010年度から2015年度まで、あまり変わってはおらず、多くの学習者は、中学校・高等学校で、プログラミングを経験せずに大学に入学している状況である。

前学習指導要領を履修した学生が入学して10年が経過し、全体として、初等中等教育では安定して一定程度の情報教育が実施されていることを示す結果であると考えられる。一方、図1の文書作成、表計算において「自信がない」との回答割合が、2010年度位をピークに、近年増加しているように見受けられ

るのが気になりである。図1の調査の際、学生には、PCや携帯電話、スマートフォン等の所持といった情報環境や、メールや電子掲示板等の使用状況、コンピュータ等の印象といった情報環境への適応状況も合わせて確認している。図2は、コンピュータの印象を、「好き」「面白そう」「難しそう」「嫌い」「その他」の5つの選択肢から、最も当てはまるものを1つ選択させた年次変化である。

2012年度から「好き」「面白そう」といった肯定的印象は減り、「難しい」との回答が増えている。図3には、メール使用に際しての情報端末機器の変化を示す。2011年度までは、パソコンと携帯電話等の両方を用いる学生が増えていたが、その後は減少に転じた。学生は、便利なスマートフォン等を利用し、パソコンの使用頻度が総じて減っている。これらが、コンピュータが「難しい」印象や、文書作成や表計算等のリテラシー能力の自信のなさに表れているようにも見受けられる。

2.2. 高等学校までのプログラミングの経験と印象

プログラミング経験とプログラミングについての印象を、文系・理系に分けた2015年度の調査結果を表1に示す。プログラミングの印象についても、前節のコンピュータの印象と同様に、「好き」「面白そう」「難しそう」「嫌い」「その他」の選択肢から最も当てはまるものを1つ選択させている。

表1. プログラミング経験の有無によるプログラミングの印象の違い（文系は全体600名弱の約15%が経験あり、理系は理系全体1800名強の約19%が経験者である）

文系	好き	面白そう	難しそう	嫌い	他
経験	6%	36%	49%	8%	1%
未経験	0.8%	16%	76%	4%	2%
理系	好き	面白そう	難しそう	嫌い	他
経験	11%	39%	42%	6%	2%
未経験	7%	26%	68%	4%	2%

表1の結果は、プログラミング経験の有無により、プログラミングに関する印象が異なることを示しており、「好き」「面白そう」との肯定的回答の割合の差を検定すると、有意であることが示された ($p < 0.01$)。つまり、プログラミングを経験することに

より、プログラミングの印象が肯定的に変化している。このプログラミング経験は、授業に関わらない個人の趣味での経験も含むため、中学校、高等学校の授業でのプログラミング経験による差を同様に検定した結果を表2に示す。

表2. 授業でのプログラミング経験の有無によるプログラミングの印象の肯定的回答の割合に関する検定

中学での授業	肯定的回答の割合		検定結果
	経験者 (対象の全体人数)	未経験者 (対象の全体人数)	
文系	29% (24)	20% (556)	有意でない
理系	44% (91)	30% (1755)	有意 ($p < 0.05$)
高校での授業	肯定的回答の割合		検定結果
	経験者 (対象の全体人数)	未経験者 (対象の全体人数)	
文系	39% (63)	18% (517)	有意 ($p < 0.01$)
理系	41% (259)	30% (1587)	有意 ($p < 0.01$)

高等学校の授業におけるプログラミング経験は、文系・理系とも、プログラミングの印象を肯定的に変化させている。中学校では、授業経験者が少ないため、文系学生に対しては、有意な結果が得られなかったが、その他は、中学校・高等学校とも、授業でプログラミングを経験した方が肯定的回答をする割合が高い。図4にプログラミングの印象を、学生の文系理系の別および高等学校の授業でのプログラミング経験の有無により分類した結果を示す。文系学生は、プログラミングの経験者では、「面白そう」との回答割合が増えており、プログラミングによる可能性を肯定的に判断しているようである。理系学生は、プログラミング経験により、「好き」「面白そう」がともに増加する一方で、「嫌い」との回答割合も増えている（授業経験者9%，同未経験者3%）。同様に検定を行ったところ、こちらも有意であった ($p < 0.01$)。つまり、高等学校の授業におけるプログラミング経験は、プログラミングの印象を全体と

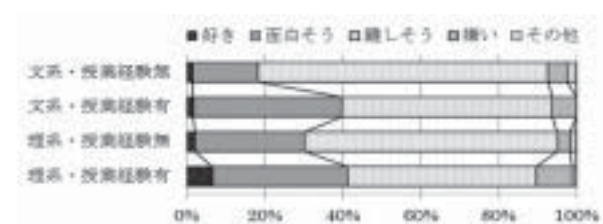


図4. 高等学校の授業でのプログラミング経験によるプログラミングの印象の違い

して肯定的にする一方、理系の一部学生に対しては、苦手意識を植え付ける状況になっているようである。なお、中学校においては授業でのプログラミング経験者が少ないため、有意な結果は得られなかった。

表3. コンピュータとプログラミングの印象の調査 (2015年度入学生に対する調査結果)

	プログラミングの印象					
コンピュータの印象		好き	面白 そう	難し そう	嫌い	その他
	好き	48	213	89	3	6
	面白そう	7	299	237	4	10
	難しそう	2	111	1143	36	11
	嫌い	1	2	63	55	3
	その他	1	21	44	6	13

プログラミングに対する印象は、コンピュータに対する印象とは一般には異なるものであるが、表3で示す通り、その関係は深い。プログラミングで肯定的な印象を持つ場合には、コンピュータ全般にも肯定的な印象を持つことが期待されるとも言える。また一方で、プログラミングを難しそうと感じてはいても、コンピュータに対して「面白そう」と回答する割合も一定程度 (237名:1割相当) いることがわかる。

2.3. 必修の情報教育におけるプログラミング教育

本学では、2006年度から一般情報教育のカリキュラムを改訂した。図5に前期必修の実習型授業である情報学Iの2015年度の授業スケジュールを示す。左列の数字が各週を示し、当該の行で、その週の学習項目が表されている。例えば4週目は、キー入力、情報倫理、情報検索、画面キャプチャ、計算処理、討論を行うといった具合である。情報学Iの授業は下記の特徴があるが、詳細は参考文献を参照されたい (布施 2011)。

- ・文系理系に依らず統一カリキュラムで授業を実施
- ・20名程度の少人数教育を実施
- ・多数の必須課題と必須要件を設け、それらを満足することを単位習得の条件とする
- ・自習対応として、他クラスの情報教学I実施時間

週	学習内容				
1		情報教育Ⅱ	基礎的なプログラミング コンピュータ環境でのプログラミング環境、ライブラリ、基本動作		
2	ケース1	プログラミングⅡ 情報教育Ⅱ	ファイルの読み込み ウィンドウ メニュー	数値データの処理	プログラムの実行
3		情報教育Ⅲ		数値データによる データの処理	評価
4		情報教育Ⅳ	領域特定	数値データによる データの処理	
5		情報教育Ⅴ		数値データによる データの処理	
6		情報教育Ⅵ		数値データによる データの処理	
7		情報教育Ⅶ		数値データによる データの処理	2週間の学習 プログラムの実行
8		情報教育Ⅷ		数値データによる データの処理	
9		情報教育Ⅸ (プログラミングⅢ) プログラミングⅢ	プログラミングⅢ Scratch	数値データによる データの処理	
10		情報教育Ⅹ	プログラミングⅣ Scratch	数値データによる データの処理	
11		情報教育Ⅺ	プログラミングⅤ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
12		情報教育Ⅻ	プログラミングⅥ Scratch	数値データによる データの処理	
13		情報教育Ⅼ	プログラミングⅦ Scratch	数値データによる データの処理	
14	実入	情報教育Ⅽ	プログラミングⅧ Scratch	数値データによる データの処理	
15		情報教育Ⅾ	プログラミングⅨ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
16		情報教育Ⅿ	プログラミングⅩ Scratch	数値データによる データの処理	
17		情報教育ⅰ	プログラミングⅪ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
18		情報教育ⅱ	プログラミングⅫ Scratch	数値データによる データの処理	
19		情報教育ⅲ	プログラミングⅬ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
20		情報教育ⅴ	プログラミングⅭ Scratch	数値データによる データの処理	
21		情報教育ⅵ	プログラミングⅮ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
22		情報教育ⅶ	プログラミングⅰ Scratch	数値データによる データの処理	
23		情報教育ⅷ	プログラミングⅱ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
24		情報教育ⅸ	プログラミングⅲ Scratch	数値データによる データの処理	
25		情報教育ⅹ	プログラミングⅴ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
26		情報教育ⅺ	プログラミングⅵ Scratch	数値データによる データの処理	
27		情報教育ⅻ	プログラミングⅶ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
28		情報教育ⅼ	プログラミングⅷ Scratch	数値データによる データの処理	
29		情報教育ⅽ	プログラミングⅸ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
30		情報教育ⅾ	プログラミングⅹ Scratch	数値データによる データの処理	
31		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
32		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
33		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
34		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
35		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
36		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
37		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
38		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
39		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
40		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
41		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
42		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
43		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
44		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
45		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
46		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
47		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
48		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
49		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
50		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
51		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
52		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
53		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
54		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
55		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
56		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
57		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
58		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
59		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
60		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
61		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
62		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
63		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
64		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
65		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
66		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
67		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
68		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
69		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
70		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
71		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
72		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
73		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
74		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
75		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
76		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
77		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
78		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
79		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
80		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
81		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
82		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
83		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
84		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
85		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
86		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
87		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
88		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
89		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
90		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
91		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
92		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
93		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
94		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
95		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
96		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
97		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
98		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	
99		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	相互評価 相互評価の成績 と最終レポートの 提出
100		情報教育ⅿ	プログラミングⅺ Scratch	数値データによる データの処理	

図5. 情報学Ⅰの2015年度授業スケジュール

帯に、TA に対する質問対応の体制を取っている等である。なお、学習者の状況に応じ、具体的な授業内容は、毎年更新を行っている。

2015 年度は、あらわにプログラミングを扱うのは、第9週で、その確認テストを第10週で行う。その他、プログラミングに関係ある授業課題としては、音の生成（第8週）、HTML 文書（第12～15週）等がある。

- ・音の生成・・・Ruby のプログラムを用いて音データを生成する。ある周波数と長さの音を生成させるサンプルプログラムを学習者に提示する。学習者は、サンプルプログラムを利用し、数値やプログラムの内容を適宜変更する等で、自らが決めたメロディに対する音データを生成させる。
- ・HTML 文書・・・HTML5 を用いた HTML 文書を各自に作成させ、相互評価を行う。加点的に JavaScript を組み込むことが可能である（必須要件とはしていない）。日本データパシフィック社による「Web 作成入門」を授業時間外に確認できるように学習者に提示している。

以下、第9週の Scratch を用いたプログラミング

の内容を示す。Scratch は2009 年度に試行的に実施し、学習者の反応が良かったことから、2011 年度からは本格的に、翌週のプログラミング小テストを含めて授業を構成している。Scratch は、MIT Media Lab の Lifelong Kindergarten Group が、8 歳以上の若年者を主たる対象として開発したプログラミング言語であり、日本では、近年、初等中等教育でのプログラミングに有望な言語としても注目されている。本授業は、文系理系に依らない統一カリキュラムのため、文系学生でも扱いやすく、プログラムの流れを確認しやすいことが求められ、それに合致している。また、サンプルプログラムが豊富で、学習者が参考にして独自の改変を行いやすい。インターフェースが工夫されており、プログラミング経験が殆どない学習者に対しても、限られた時間での実施に優れていると考えた。先の一般情報教育の実態調査においても、プログラミングを扱う授業科目のうちの3%が Scratch を取り入れていることが示されている（岡部 2014）。

学習目的は、授業全体における位置づけや時間等の制約に依存して決定される。本授業のプログラミングでは、テストも含め、高々2週程度で扱うものである。また、前述の通り、8割以上の学習者は現状としてプログラミングを経験しておらず、本授業以外では、今後も経験することがないまま社会に出る可能性もある。そのため、本授業では、まずはプログラミング体験として位置付けることにした。既存のサンプルプログラムをそのまま実行できる形で学習者に提示し、指導者は、そのプログラム内容を端的に説明する。学習者はその実行手順を学び、実際にサンプルプログラムを実行させるとともに、内容を修正・再実行させ、結果を確認していく。簡単なサンプルプログラムであるが、コンピュータ上のプログラミング実行による処理結果を実際に自身で体験・確認することで、コンピュータ上での自動処理の可能性に思いを至らすきっかけにはなるように意図している。翌週の小テストでは、プログラムの画像を目視で確認することで、プログラムの処理の推定を行わせ、学習者の理解度を把握することとした。

授業では、時間的制約で高々1コマ20分程度しか、直接の指導はできない。学習者は、最終成果物

を自習の中で完成させる。不明点は、授業の空き時間に直接 TA に質問することもできる。課題は、サンプルを元にしたプログラムの修正、または独自プログラムを作成し、それら学習者自身のプログラムの説明をレポートとしてまとめて提出する。サンプルプログラムを修正した場合には、どのサンプルをどのように修正したかを記載し、独自プログラムの場合には、どのようなプログラムを作成したかを、工夫点を含め記載する。

翌週には 10-15 分程度の小テストを実施し、特に判断と繰り返しについての理解を確認することとした。



図 6. Scratch のサンプルプログラム画面

Scratch のサンプルプログラムの画面を図 6 に示す。左側のカテゴリ（制御、行動など）から適当なブロックを選択し、中央のスクリプト部分に配置し組み合わせることで、プログラムを作成する。また、図 7 に、プログラミング小テストの設問例を示す。プログラム画面を画像として表示し、実行時の動作を推定させる形式としている。



図 7. プログラミング小テストの設問例

なお、毎週授業時に行わせている作業記録によると、2015 年度の Scratch プログラミングの授業から翌週までの間の自習時間は、0（なし） 1%, 1 時間未満 4%, 1 時間以上 3 時間未満 30%, 3 時間以

上 5 時間未満 40%, 5 時間以上 25% という結果となっており、当該年度 15 回の授業の中で最も自習時間が多い週となっていた。

3. Scratch を用いたプログラミングの結果分析

3.1. 授業評価から見る Scratch の印象と自己評価

本節では、まず、2015 年度の授業最終日に実施した授業評価におけるプログラミング関連項目の結果を確認する。回答割合は 7 割弱、回答数は約 1700 であり、Scratch に関しては以下の設問を行い、1)~4) の設問の平均値と標準偏差（SD と表記）の値は表 4 に記載した。

1) Scratch を学ぶことの難易度（7 段階）

1：全く難しくない—4：普通—7：とても難しい

2) Scratch の面白さの程度（7 段階）

1：全く面白くない—4：普通—7：とても面白い

3) 提出した Scratch プログラムの自己評価（7 段階）

1：低い（不満）—4：普通—7：高い（満足）

4) Scratch は大学の一般情報教育でのプログラミング教育に適切か。（7 段階）

1：全く適切でない—4：普通—7：とても適切

5) Scratch を利用したプログラミングの授業は、どの校種に適用したら最もよいと考えるか

1：小学校—2：中学校—3：高等学校—4：大学

表 4. Scratch の印象と提出プログラムの自己評価

		難易度	面白さ	作品の自己評価	大学教育への適用
文系	平均	4.7	4.3	3.8	4.0
	SD	1.51	1.65	1.46	1.39
理系	平均	4.6	4.5	3.9	4.0
	SD	1.61	1.67	1.50	1.48

全体に文系・理系とも平均値は似ているが、難易度と大学教育への適用へのばらつきは理系が大きいようである。5) の望ましい校種については、高等学校を上げる学習者が文系（38%）・理系（35%）とも最も多く、続いて文系は大学を、理系は中学校を挙

げていた。

次に、学習者の Scratch に関する難易度と面白さとの関係を表5に、提出した自身のプログラムに関する自己評価と、難易度、面白さの回答との関係を表6に示す。見やすさのために、各項目を肯定的、普通、否定的といった3段階に分類しなおし、各対象者を母数とした割合で表記した。最も値が高い割合に網掛けを施している。

表5. 文系理系の別による面白さと難易度の関係

難易度	面白さ					
	文系			理系		
	面白くない	普通	面白い	面白くない	普通	面白い
難しく ない	3.7%	3.3%	9.8%	4.4%	3.3%	11.8%
普通	4.7%	9.6%	12.8%	3.5%	9.8%	13.2%
難しい	17.9%	16.8%	21.4%	14.3%	15.2%	24.6%

表6. 課題として提出したプログラムの自己評価の高低と難易度・面白さの関係

難易度	提出したプログラムの自己評価					
	文系			理系		
	低	普通	高	低	普通	高
難しく ない	3.7%	4.9%	8.2%	4.4%	6.2%	8.9%
普通	7.7%	12.1%	7.2%	5.9%	11.6%	8.9%
難しい	27.5%	18.6%	10.0%	24.4%	17.4%	12.3%
面白さ	提出したプログラムの自己評価					
	文系			理系		
	低	普通	高	低	普通	高
面白く ない	17.0%	6.3%	3.0%	11.9%	7.0%	3.3%
普通	10.7%	13.8%	5.1%	10.6%	12.8%	5.0%
面白い	11.2%	15.6%	17.2%	12.1%	15.4%	21.8%

表5より、Scratch によるプログラミングは、面白く、かつ難しいとの回答が、文系・理系とも最も多いことが示されている。森らの小学生を対象とした Scratch を用いたプログラミングによる先行研究でも、楽しさと簡単さの観点から評価を行っており、学習者は、楽しさは高評価であるが必ずしも簡単であるとは回答しておらず、それと同じ傾向と言える(森 2011)。表6では、課題として提出したプログラムの自己評価の高低を示したが、自己評価が低い学習者は、殆どが Scratch を難しいと回答している

ことがわかる。一方、面白いと回答した学習者は、自己評価が高い割合が多いが、文系では、同程度の割合で「面白くなく自己評価も低い」ものが見受けられる。表5の「面白くなく難しい」と回答した学習者に対応すると考えられ、この割合を減少させることが重要であるとする。

3.2. プログラミング小テストの結果と分析

本項では、小テストの結果とその分析を行う。2015年度は、学習者の履修した曜日講時により、2種類の小テストを実施した。表7はその抜粋である。

表7. 小テストの出題パターン(抜粋)

	A パターン	B パターン	選択肢
設問1	6回繰り返す 100 歩動かす 60度回す すべてを止める (本例は図7参照)	6回繰り返す 100 歩動かす 120度回す すべてを止める	三角形 四角形 五角形 六角形 星形
設問2	5 回繰り返す 2 回繰り返す 4 歩動かす すべてを止める	4 回繰り返す 3 回繰り返す 5 歩動かす すべてを止める	適当な 歩数の 5 択
設問3	ずっと 100 歩動かす 90度回す のプログラムは四角形を描く。これを変更し、三角形を描くためにはどうするとよいか	ずっと 100 歩動かす 90度回す のプログラムは四角形を描く。これを変更し、六角形を描くためにはどうするとよいか	後述するが、 選択肢は5択

表8に、A、B 各パターンを適用したあるクラスの正答率の結果を示す。実は、A と B の出題に対する解答パターンには、大きな違いがある。まず、設問1における注目すべき数値は、繰り返しの「6回」と、回転の「60度」「120度」である。A パターンでは、6回の繰り返しから正答である「六角形」を選択する学習者(思考としては誤りであるが回答は正解となる)、「60度」回転から正答である「六角形」を選択する学習者(正答)、「60度」回転から「三角形」を選択する学習者(誤答)に大きく分けられる。一方、A パターンの設問3では、回すブロックの90度を120度に変える正答と、回すブロックの90度を60度に変える誤答を5つの選択肢のうちの2つとして用意したところ、120度に変えると正答した者

の殆どは、設問1をほぼ全て正解し、60度に変える誤答をした学習者は、設問1での正誤が分かれていることがわかった（表9参照）。

表8. プログラミング小テストの正答率の違い

	A パターン出題		B パターン出題	
	文系 (165 名)	理系 (395 名)	文系 (184 名)	理系 (261 名)
設問 1	69%	73%	16%	30%
設問 2	99%	100%	98%	98%
設問 3	28%	52%	22%	36%

表9. A パターンの主な正誤回答数（正答は網掛け）

設問 1	設問 3			
		90 度→120 度	90 度→60 度	
	三角形	0 (文) / 1 (理)	42 (文) / 94 (理)	
	六角形	47 (文) / 199 (理)	58 (文) / 59 (理)	

表9より、Aパターンでは、スプライト（猫オブジェクト）の進行方向からの回転角度ではなく、描く図形の内角を回転の際に表記するという誤った認識をしている学習者がいることがわかる。また、設問1で、繰り返しの回数等を鑑みて、安直に六角形と回答した学習者も一定割合存在することがわかる。

一方、Bパターンでは、設問1の6回繰り返しはダミーで、実行結果は、三角形を2回重ね描きする。Aパターンに比して設問の難易度が高い。この問題でも、120度という角度がスプライトの進行方向との角度であることを理解していれば解けるはずであるが、理系の学習者ですら3割、文系学習者では16%しか正答していない。これは、六角形の内角120度を表示すると考えている学習者や、6回の繰り返しの数から六角形と思い込む学習者が多いことを示している。Bパターンでは、設問1の正答者は、正しい認識をしていることが予想され、同様の考え方で解ける設問3も正答している割合が高い。Bパターンの結果を表10に示す。

結果として、A・Bいずれの場合においても、誤答パターンは同じ思考の誤りから発生していると考えられる。文系・理系を問わず、Scratchにおける回転の命令の正しい認識を適切に説明するような、自習でも確認可能な説明教材が望まれる。

表10. B パターンの主な正誤回答数（正答は網掛け）

設問 1	設問 3			
		90 度→120 度	90 度→60 度	
	三角形	27 (文) / 73 (理)	3 (文) / 6 (理)	
	六角形	6 (文) / 16 (理)	128 (文) / 157 (理)	

4. 考察

前述の通り、本学では一般教育としての情報教育において、Scratchを用いたプログラミング体験を行っているが、難しく面白いという学習者の認識が多数であることがわかる。また、小テストの結果からは、正確な理解をしている学習者は、残念ながらさほど多くはないことが示された。その意味でも、現状はプログラミングを体験する学習にとどまっていると言える。一方、情報学Iでは、音のデジタル化の学習で、独自の音データを生成させる際に、ツールとしてプログラムを用いている。当該課題は、プログラミングを行うことが主目的ではないが、当該課題の達成に際し、その前にプログラミングの学習を行ったか否かで、理解度に差が生じていることが経験的に判明している。つまり、本授業のような短時間の体験学習でも、プログラミングを経験し、プログラムにおける判断や繰り返しの概念を把握することは意味があるとの結果であると思われる。

授業評価において、色や音のデジタル表現における関心度と、Scratchの面白さとの関係、ならびに、Scratchの面白さと、授業全体における「授業により知的に刺激され、さらに深く勉強したくなった」との設問との関係性をそれぞれ表11、表12に示す。

表11. Scratchの面白さと色・音の内容の関心度

		色・音のデジタル表現の関心度		
		肯定的	普通	否定的
文系	面白くない	6.3%	4.4%	15.7%
	普通	15.5%	7.7%	6.1%
	面白い	24.8%	7.5%	11.9%
理系	面白くない	7.6%	4.1%	10.4%
	普通	15.9%	6.3%	6.2%
	面白い	39.9%	4.1%	5.5%

表 12. Scratch の面白さと知的な刺激や深い勉学意欲

		知的な刺激と深い勉学意欲		
		肯定的	普通	否定的
文系	面白くない	6.6%	8.1%	11.8%
	普通	10.7%	11.8%	7.1%
	面白い	25.8%	11.4%	6.6%
理系	面白くない	5.6%	7.9%	8.5%
	普通	11.4%	11.6%	5.5%
	面白い	30.7%	13.8%	4.9%

表 11, 12 とも, Scratch を面白いと回答した学習者は, 色・音のデジタル表現でも, 授業全体における知的な刺激としても高い肯定感を持っていることがわかる。授業評価において, 本授業で習得できたことを自由記述で記載させたが, Scratch をコメントしている学習者は合わせて HTML や音についても言及していることが多い。以下に一例を示す。

- ・今まで苦手だったエクセルやワードが簡単にできるようになった。また, 音やプログラミングや HTML の課題が面白く, 実はこの方面も好きだったと気付いてよかった。(理系)
- ・いろいろである。例えば, HTML 文書や Scratch ははじめて習ったことで, すごく楽しかったとおもう。それ以外に, 情報倫理に関する知識も得た。(理系)
- ・html 言語が面白く, 自分でもいろいろ作ってみたいと思うきっかけとなった。今後の進路選択でも参考になる。Scratch に関しても, わからないことはいまだに多いが, プログラミングのイメージをつかむにはとても良いと思う。また, 全体を通してエクセルの使い方が身についたのが普段自分の日常生活においてもっとも役に立つかもしれない。(理系)
- ・情報倫理に関する知識はもちろんのこと, 個人的にはスクラッチのプログラミングや HTML 言語の習得ができて非常にためになった。自分自身でも, これらに触れる機会を多くしていきたい。(文系)

初等中等段階で, 一定以上の時数を用いたプログラミング学習が行われた場合, それを前提として, 大学における一般情報教育における各学習項目を, より高度に構成し, 展開していくことが可能となる

と考える。

5. まとめ

本論文では, 本学における一般教育としての情報教育において, プログラミングをどのように取り上げているかを含めた実践内容について報告し, 実践結果に基づく考察を行った。現状では, Scratch を用いてのプログラミング体験に留まっており, 小テストにおける学習者の理解度も高いとは言えない。今後は, まずは, 小テストにおける誤りの要因を踏まえた補助教材を拡充させることで, 学習者の認識を改善することを検討する。その過程を通し, 面白いが難しいとの回答が多いプログラミングの印象が改善するか否かを確認していきたいと考える。また, 情報学 I の各学習項目に分散させたプログラミングの要素を組み合わせ, 学習者の知識を構築していく学習構成についても検討していきたい。例えば, 現状では, Scratch のプログラミングの他, Ruby や JavaScript をツールとして用いて, 課題解決を行っている。しかし, それぞれが個々の学習項目に閉じている状況である。各学習項目におけるプログラムの内容を整理統合して学習者に提示することにより, プログラミング全体に関する認識を深化させることも可能になると思われる。

近年の情報技術の進展の中, 一般教育におけるプログラミング教育は, 初等中等教育課程を含め, 全学習者に対するプログラミングを体験から始まり, その経験を踏まえ, 問題解決を行うツールとしての活用や, プログラムを用いての創作的活動といった表現の実現手法としても考えられると思われる。初等中等教育段階におけるプログラミング教育の進展に応じ, 大学入学時の学習者の状況が刻々と変化していくであろうことが予想されるため, 学習者の状況を把握しながら, 一般情報教育の個々の学習内容の理解を深化させるためにプログラミングの学習を随時組み合わせ活用していくことを考えていきたい。

参考文献

- 1) National curriculum in England: computing programmes of study, Published 11 September 2013, <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>, (2017 年 6 月 17 日閲覧)
- 2) 「世界最先端 IT 国家創造宣言」の変更について, 平成 27 年 6 月 30 日閣議決定, <https://www.kantei.go.jp/jp/singi/it2/kettei/pdf/20150630/siryoul.pdf>, (2017 年 8 月 29 日閲覧)
- 岡部成玄 (2014), 「一般情報教育の全国実態調査」, 『情報処理学会誌』 55 (12), 1400-1403 ; 56 (1), 94-97
- 布施泉, 岡部成玄 (2011), 「北海道大学における全学教育としての情報教育」, 『情報処理学会誌』 52 (10), 1341-1345
- 森秀樹, 杉澤学, 張海, 前迫孝憲 (2011), 「Scratch を用いた小学校プログラミング授業の実践」, 日本教育工学会論文誌 34 (4), 387-394