

大学の一般情報教育の中でのプログラミング教育の実践

布施 泉*1・岡部成玄*1
Email: ifuse@iic.hokudai.ac.jp

*1: 北海道大学

◎Key Words 大学, 一般教育, プログラミング

1. はじめに

近年, 初等中等教育段階におけるコンピューティング教育が世界的に推進されている。例えば英国では, 2014年9月より, 小学校からのコンピューティング教育を必修で実施している。すべての子供たちに対し, 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する経験を積み重ねる学習を行うものであり, 技術者育成のための教育ではない¹⁾。我が国でも, プログラミング教育の充実が政府のIT戦略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。

この10年で, 社会の情報化は大きく進展し, 子どもたちは幼い時から, 身近にコンピュータを有した社会の中で育っている。ほぼ全員が, スマートフォンを持って大学に入学している昨今, 大学における一般教育としての情報教育の中で, プログラミング教育をどのように位置づけるのが適切であるか, 学習者の状況を踏まえながら検討を行う必要があると考える。

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実態調査を行っており, 情報処理学会誌でその結果が報告されている³⁾。全国の対象大学の3割強が回答した結果であり, そこでは, 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ているとある。選択を含めたすべての回答科目全体の中で, 約1割が科目全体としてプログラミングを取り上げている。また, 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, 全体の3割程度である。すべての大学生が履修する一般科目として, 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と言え, プログラミングに全く触れることなく, 大学を卒業する学生が一定以上存在する可能性が高いと考えられる。

このような背景の中, 本稿では, 北海道大学における一般情報教育におけるプログラミング教育の実践について報告する。北海道大学では, 1年生を対象とする一般情報教育科目を2科目開講している。前期は必修2単位で主に実習を行う科目(情報学I), 後期は選択2単位で講義(および実習)を行う科目(情報学II)である。ともに, 広く情報科学および情報社会に関わる内容を含む科目であるため, プログラミングのみを科目全体で行うことはできない。また, 必修の授業は, 理系文系を問わず統一的内容で行っているため, 文系でも扱いやすいプログラミング言語を取り上げることが必須となる。近年は, MITメディアラボが開発したプログラミング言語であるScratch⁴⁾を用いて授業を行

っている。一方, 後期の選択の情報科目は, 担当教員の自由度が比較的高く, 近年著者らは, 次のプログラミング言語(Scratch, JavaScript, Sunaba⁵⁾)を用いて授業を行ってきた。本稿では, これらのプログラミングの学習において著者らが行った実践事例を, 学習者の反応を含め報告する。

2. 大学入学時の調査

一般に授業を設計する際には, 学習者の状況を調査することが必要である。必修の情報学Iの第一回授業で, 学習者の状況について, 調査を継続的に行っている。本稿では, その中で, プログラミング教育の設計に関連する調査結果の概要を示す。北海道大学では毎年2600名程度の学生が入学しており, 原則として, 設問の対象者数は, 各年度全体では2600名程度である。

2.1 学習者の情報環境

図1は学習者のプログラミング経験についての回答結果である。図中の「旧」「新」の表記は, それぞれ旧課程, 新課程を履修した学生を示す。また, 「現」の表記は, 当該年度の現役生の回答結果を示す。回答は, 「できる, している」「経験がある」「経験がない」の3択であるが, 図では, 「経験がある」または「できる, している」と回答したものをA, 「経験がない」と回答したものをCと表記した。入学生のプログラミング経験は, 徐々に増えているが, 2016年度の新課程者においても約2割に留まっており, 現状では, 大学入学時にはプログラミングを習った学生は少ない状況である。

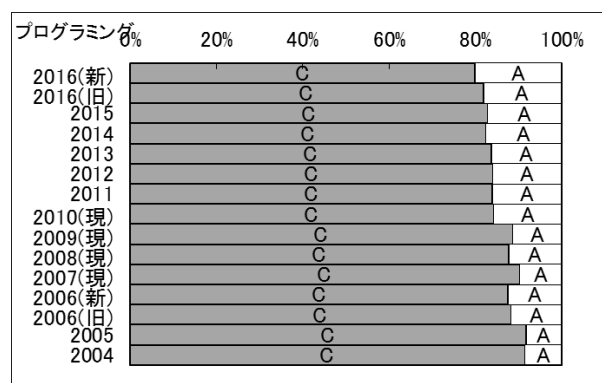


図1 大学入学時におけるプログラミング経験

大学生は, 日常的にはスマートフォンを用いて情報を取得していることが多いと考えられる。パーソナルコンピュータの所持について, 自分専用か否か, 使っ

ているか否か、の組み合わせに加え、持っていないを選択肢として、計5択で訊いた回答結果を図2に示す。2016年度は、持っていないと回答する割合が前年度までに比較して高く、13%に達する程度である。

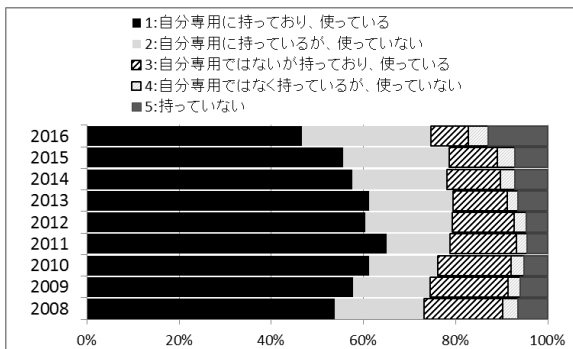


図2 パーソナルコンピュータを所持する割合

2.2 プログラミングに対する意識

プログラミングに対する学習者の意識を図3に示す。プログラミングの印象を、「好き」「面白そう」「難しそう」「嫌い」「その他」の5択で訊いたもので、2010年からの結果を示す。図3の上図は、図1のAに対応する「できる、している」「経験がある」と回答した学習者を対象としたものであり、下図は図1のCに対応する「経験がない」と回答した学習者のものである。

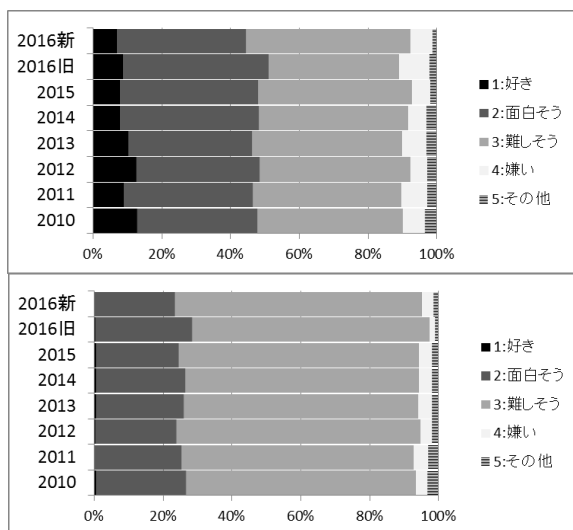


図3 プログラミング経験とプログラミングの印象

プログラミングを経験していない場合、プログラミングに対し、難しそうという印象が高いことがわかる。一方で、プログラミング経験者は、「好き」「面白そう」といった肯定派と「難しいそう」「嫌い」といった否定派がほぼ半々である。

「はじめに」で述べた通り、今後、初等中等教育段階で、プログラミング教育が推進されていくと考えられるが、上記の通り、現状の大学入学生の状況を鑑みると、約8割の学習者はプログラミングを経験していない。さらに、その未経験者は、プログラミングに対して、「難しそう」と言った苦手意識を持っている割合が、7割程度と高いことが示された。

3. 大学一般教育におけるプログラミング教育

3.1 プログラミング教育の位置づけ

前章の状況を踏まえ、現状では、まずは、学習者の苦手意識等を意識しない程度に敷居を下げた上で、プログラミングの導入をまずは進める必要があると考える。さらに、プログラミングの経験をしたのちは、学習者の習得状況を踏まえ、各科目の到達目標を設定することが必要であると考えられる。

3.2 必修授業におけるScratchを用いたプログラミング教育の実践

必修の情報学Iの授業では、Scratchを用いたプログラミングを行っている。授業時間内では、1コマの中の一部の時間でサンプルプログラムの説明をする。当該授業における課題として、サンプルプログラムの修正もしくはオリジナルプログラムの作成を課す。学習者は主として、授業時間外に自習として課題を行う。また、学習者には、翌週、条件分岐と繰り返しの理解を確認するためのプログラミング小テストを課している。



図4 2つのオブジェクトにおける処理の例

図4は、2016年度の授業テキストに記載したプログラム例である。授業では、それぞれのオブジェクト毎にプログラムが記述できることを示し、オブジェクト間のインタラクティブな動作が可能とする中で、最低限の条件分岐と繰り返しを理解することを目的とするものである。

3.3 選択授業・情報学IIにおけるScratchを用いたプログラミング教育の実践

情報学Iでは、学習者は全員、Scratchのプログラミングを行う。その上で、後期に選択科目としての情報学IIを履修することになる。前述したとおり、後期の授業は、統一カリキュラムでは行ってはいないため、プログラミング言語の選択は、担当教員の裁量に任されている。本節では、前期のScratchをそのまま用い

る授業実践について述べる。これは、第一著者が 2014 年度後期等で行った実践例である。前期の情報学 I では、課題としては、サンプルプログラムを修正でも可としているが、後期では、ある指定された設定で問題解決を図ることを目標とし、迷路脱出を課題とした。本実践においては、授業は 1 コマ 90 分を用いた。標準の第一課題は、指定されたフィールド (図 5 参照) において、オブジェクトを、上下左右キーを使って動かし、学習者が操作する中でゴールを目指すものである。さらに当該課題を達成したものは、加点課題として、迷路の自動脱出に挑む。標準課題は、ほぼ全員が授業時でクリアをしている状況である。

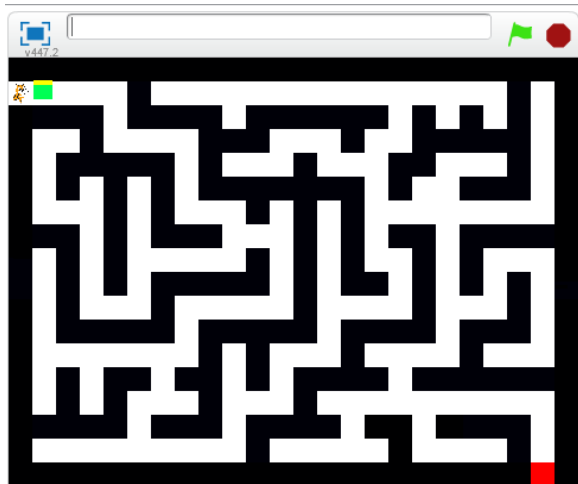


図 5 迷路脱出の設定例 (猫のオブジェクトは手動で脱出させ、四角オブジェクトは自動脱出をさせるように指示している)

3.4 選択授業・情報学 II における JavaScript を用いたプログラミング教育の実践

本節では、第二著者が 2015 年度に行ったプログラミング教育について報告する。使用言語は JavaScript であり、前期とは別の言語で、直接プログラムの命令をテキストで記載する必要があるため、学習者にとっては比較的敷居が高いと考えられる。一方で、JavaScript は、HTML の中で動的処理には不可欠なものである。また、メモ帳などのテキストエディタでプログラムの作成ができ、ウェブブラウザで実行することができるため、学習者は、特別な環境を用意する必要がないことが大きなメリットである。また、汎用性が高いため、興味を持った学習者がさらに学習を進めるために適した言語であるとも考える。

授業時には、学習者に対して、具体的な操作と実行方法を示し、試させるとともに、さまざまなサンプルプログラムを資料とともに提示する。課題としては、まずは学習者に対し、実際にサンプルプログラムを実行すること、サンプルプログラムの一部について、変更を加え、実行させ、結果と考察の報告をすること、を課した。具体的なサンプルプログラムの一例を図 6 に示す。学習者には、階乗の計算を行い、ウェブブラウザ上に結果を静的に表示する例のほか、ライフゲームやハノイの塔 (並べ替え) 等、時間進展とともに描画を行うものなどをサンプルとして提示した。



図 6 階乗計算とライフゲームの例

3.5 選択授業・情報学 II における Sunaba を用いたプログラミング教育の実践

本節では、第一著者が 2015 年度に行ったプログラミング教育について報告する。使用言語は、Sunaba⁵⁾とした。主に以下の理由による。

- ・覚えるべき文法が少ない
- ・プログラムの記述は、テキストエディタで行える
- ・実行が簡単である
- ・関数を作ることができる。それを用い、他者のプログラムを部分的に取り入れて、プログラムを組み立てることも可能である

選択した情報科目ではあるが、学習者はプログラミングに対し苦手意識を持つ人が少なくない。プログラム全てを自分で記述して実行することに対する敷居が高い学習者が一定以上いることが予想されたため、課題としては、プログラムが得意な者が他者を助けることを可能とするような取り組みを視野に入れた。



プログラムはテキストエディタで作り、この画面にドラッグする。

プログラムの実行結果の表示画面 100×100の画素の色等を指定する。

図 7 Sunaba でのプログラミング課題の説明

図 7 の画面右で示した通り、プログラムの実行結果は、100×100 ピクセルの画像となる。学習者は各ピクセルの描画を、プログラムを実行することで行う。具体的な課題は、100×100 ピクセルの絵を描くことであり、プログラムファイル (テキストファイル) と、何を行ったかを説明するファイルを合わせて提出させた。また、プログラムは、繰り返しの処理を必須とすることとした。学習者間の助け合いを推進するために、フォーラムを用意し、自身が作ったプログラムは、そのフォーラム内であれば、他者に提供してもよいこととした。一方で、実際に他者のプログラムを使用する場

合には、プログラム中にコメント文で、誰の何のプログラムを使用したかを記載させた。

Sunaba における計算式の処理手順が少々、通常の数学的処理とは異なる箇所があり、プログラムの実行がなかなか上手くいかなかった学習者がいた。また、Sunaba では、整数値のみで値を表現するため、簡単ではあるものの、誤解をして上手く実行できないプログラムを作る学習者もいた。

4. 学習者の反応等の考察

3章で示した通り、本カリキュラムにおいては、一般教育としての情報教育の中では、多くても数回しかプログラミングの時間をとることができない。その中で、まずは学習者の敷居を下げることを目的に、必修授業において、Scratch を全学習者に行わせることとした。簡単なプログラミングの授業ではあるが、全体として、学習者のプログラミングに対する敷居を下げることに役立っていると考えられる。

表1は、必修の科目である情報学Iの2015年度の授業評価における、プログラミング言語 Scratch の印象と、自身の提出した提出プログラムの自己評価を行った結果を示した表である。難易度、面白さ、作品の自己評価、大学の一般情報教育で Scratch を取り入れたプログラミングを行う適切性、を7段階で評価させた。値が高い場合、難しさ、面白さ、高自己評価、適切性、の程度がそれぞれ上がるような指標である。表1の通り、文系理系に関わらず、難易度、面白さ、作品の自己評価、大学教育の適用は、平均値としてはほぼ同等であり、通常の評価を得ていることがわかる。

表1 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

後期の選択のプログラミング授業においては、学習者がパソコンをあまり使わなくなり、苦手意識が表出している現状を鑑み、前期と同じプログラミング言語を選択した例、汎用的なプログラミング言語を選択した例、および初心者用の文法が少ないプログラミング言語を扱った例を示した。前期と同じプログラミング言語を用いる場合には、目標設定を少々高く設定し、問題解決のためのプログラミングを行わせることとしたが、授業時間で、標準の課題は到達できる程度の難易度であるため、学習者の負担は少ないように感じられる。JavaScript の例では、実用に供しているプログラミング言語であり、興味のある学習者には、深い学びへの導入となりそうである。課題設定としては、初心者配慮してプログラムの実行を主とした体験型の授業であるため、学習者の負担はさほど高くないように思われる。最後に、初心者用プログラミング言語 Sunaba では、これは実践的に利用した2つのクラスで別々の状況となった。さまざまな学習者が、フォー

ムに投稿してくれた積極的なクラスでは、課題の質もよく、学習者もさほど不満がなかったようであるが、フォーラムへの投稿がほとんど行われなかったクラスでは、学習者の負担が高くなり、一部、難易度に関する不満も表出した。

5. おわりに

本稿では、大学入学時のプログラミングに関する状況を踏まえ、著者らが北海道大学の一般情報教育の授業として行ったプログラミングの実践事例を示した。プログラミング教育は、参考文献(2)の閣議決定で、「初等・中等教育段階におけるプログラミングに関する教育の充実に努め、IT に対する興味を育むとともに、IT を活用して多様化する課題に創造的に取り組む力を育成することが重要であり、このための取組を強化する」と記されており、今後、初等中等教育段階で推進されることが予想される。図3で示した通り、大学入学時の段階で、プログラミング経験のある学習者は、そうでない学習者に比して、プログラミングに対する肯定的意識が高い。しかしながら、その割合自身は現状では全体として高くはない。学習者の興味関心を引きながら、初等中等教育段階でのプログラミング教育が今後、推進されていけば、大学の一般情報教育においても、より高度で実用に供するプログラミング教育を行うことができる可能性がある。情報系に進むのではない一般の大学生に課すべきプログラミング教育については、今後も学習者の情報環境や初等中等教育での学習状況を引き続き把握しながら、授業を検討していくことが必要であると考えられる。

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>, (2016年6月15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/siryou1.pdf>, (2016年6月15日閲覧)。
- (3) 岡部成玄: “一般情報教育の全国実態調査”, 『情報処理学会誌』, 55 (12), pp.1400-1403 (2014); 56 (1), pp.94-97 (2015)
- (4) MIT Scratch Team : Scratch, <https://scratch.mit.edu/>, (2016年5月22日閲覧)。
- (5) 平山 尚: プログラミング言語 Sunaba, <http://www.page.sannet.ne.jp/hirasho/sunaba/>, (2016年5月22日閲覧)。