

共通教育におけるプログラミング入門講義

石黒 克也, 佐々木 正人, 佐々 浩司

高知大学 学術情報基盤図書館

ishiguro@kochi-u.ac.jp

Introductory lecture on programming in common education

Katsuya Ishiguro, Masato Sasaki, Koji Sassa

Library and Information Technology, Kochi Univ.

概要

高知大学の共通教育において昨年度より「初等プログラミング入門」を開講した。主な対象はプログラミング経験のない学生とし、プログラミング言語には入門用言語である Sunaba を用いた。授業の最終目標を簡易的なテトリス作成とし、15回の授業で一通りの機能を備えられるように進めた。本稿ではその実践内容を報告する。

1 はじめに

初等中等教育におけるプログラミング教育の必修化が決まり[1]、小学校は 2020 年、中学校は 2021 年、高等学校では 2022 年から新しい学習指導要領が実施される。また、2017 年には初等中等教育におけるプログラミング教育等の IT・データ教育の実装[2]も閣議決定されている。総務省でも「若年層に対するプログラミング教育の普及推進事業」[3]が 2016 年度から展開されており、小中高を含むすべての若年層を対象として、プログラミング教育を通じた ICT に関する基礎知識・スキルの定着が図られている。政府はプログラミング教育の充実を IT 戦略の一つとして位置づけており、今後ますます推進されることが予想される。

一方、大学におけるプログラミング教育の実情についてはどうか？これについては、2013 年度に情報処理学会一般情報教育委員会により行われた一般情報教育に関する調査[4]がある。対象大学の 3 割あまりから回答があり、それらの大学で開講されている一般情報教育に関する科目のうち科目全体でプログラミング教育を行っているのは 1 割ほどとのことである。高知大学(以下、本学と記す。)の共通教育でも事情は同じで、プログラミングを科目全体で行う授業は皆無、あっても情報処理の授業で数回触れるのみである。そのため、一部の学生は専門科目でプログラミングを学ぶものの、その他の学生はプログラミングに触れる機会がないまま卒業してしまう。しかしなが

ら、初等中等教育におけるプログラミング教育の必修化を見据えると、大学におけるプログラミング教育も再検討すべき時期に差し掛かっていると思われる。つまり、今後はプログラミング教育が必修化された学習課程を終えた学生が大学に入学してくるようになるため、大学でもそれに対応した学習環境や教育内容を準備しておく必要があると考えられる。

本学では昨年度よりセンター系教員の担当授業数を増やすことになったため、それを機会に全学部、全学年を対象とした「初等プログラミング入門」を開講し、共通教育においてプログラミング教育を行うことにした。プログラミング学習環境については、本学では平成 9 年度より学生にパソコンを必携させており、またどの教室でも無線 LAN につながることが可能なため、部屋や設備を特別に準備することなく PC を用いた授業を実施することが可能となっている。また本学では、全 1 年生に対して第 1 学期に共通教育で情報処理の授業を必修化している。情報処理の授業では主に大学での PC の利用方法(ネットワーク接続、メールなど)、情報セキュリティ、情報倫理、ワード、エクセル、パワーポイントなどの一般的な使い方が教えられている。授業後のアンケート調査において、情報処理の後にさらに学びたいことの上位にはエクセル、ワード、パワーポイントなどが目立つものの、それらに次ぐものとしてプログラミングもあがっている。つまり、プログラミングを学びたいという潜在的な需要があるものと考えられ

る。そのことも踏まえ、本講義の位置付けを共通教育の情報処理に続くプログラミング未経験者向けとした。小学校新学習指導要領（2017年3月公示）[5]には「次の学習活動を計画的に実施すること。児童がプログラミングを体験しながら、コンピュータに意図した処理を行わせるために必要な論理的な思考力を身に着けるための学習活動」とあるが、本講義の目的もちょうどこれに当たるようなものとなっている。プログラミング未経験者が共通教育で学ぶ際には、プログラミングの技術そのものではなく、プログラミングの考え方を身につけることを重視したほうがよいと考えてのことである。

本稿では、本学において昨年度より開講している「初等プログラミング入門」で実施した内容、アンケート結果を基に、共通教育におけるプログラミング入門講義についての実践報告を行う。

2 受講生と初回アンケート

2.1 受講生

本学の学部構成は人文社会科学部、教育学部、理工学部（旧理学部）、医学部、農林海洋学部（旧農学部）、地域協働学部の6学部からなっている。本講義は第1学期に週1回、90分授業で15回（テスト期間は除く）開講される。

2016年度の受講者数は50人で学部ごとの内訳は図1に、また2017年度の受講者数は47人でその内訳を図2に示す。

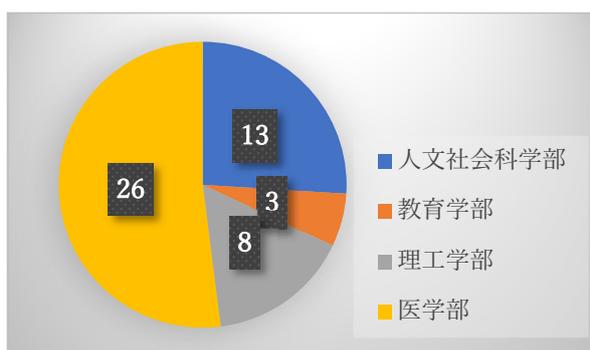


図1 2016年度学部別受講者数 (全50人)

各年度ともに1年生から4年生まで受講しているが、共通教育科目であるため1年生の割合が大きい。また、離れたキャンパスにある医学部の学生が共通教育を受講しに来る曜日に開講しているため、医学部生が全体の半数以上を占めている。

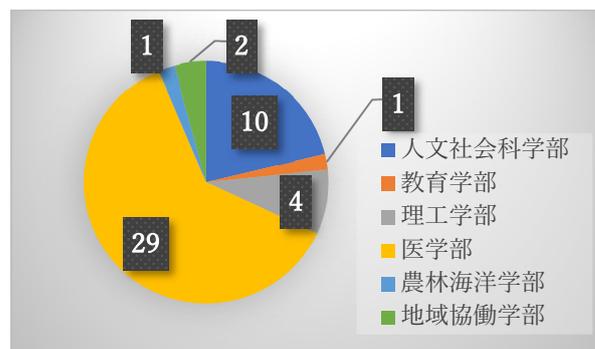


図2 2017年度学部別受講者数 (全47人)

2.2 初回アンケート

各年度の最初の授業では初回アンケートを実施し、使っているPCのOSやこれまでのプログラミング経験について統計を取った。表2は2016年度、2017年度に学生が使用していたOSの割合である。

表1 使っているOS

	2016年度	2017年度
Windows10	74%	82%
Windows8, 8.1	13%	10%
Windows7	7%	5%
MacOS	6%	3%

図3は2016年度受講者のこれまでのプログラミング経験の割合を示したものである。

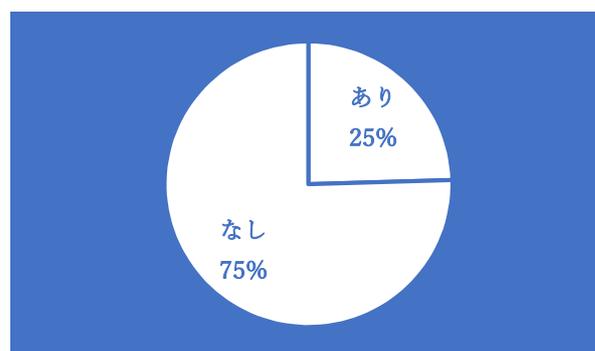


図3 これまでのプログラミング経験について (2016年度)

プログラミング経験者が使ったことのある言語にはC、C++、JavaScript、Basic、Visual Basic、Javaなどが挙げられており、授業を進めていくうちにC++を使いこなす経験者がいることも判明した。

本授業を受講した理由も尋ねてみたところ、「プログラミングに興味がある」、「プログラミングをしてみたかった」、「パソコンを活用するため

の知識を身につけたい」、「将来必要」、「将来役に立ちそう」などがあげられた。

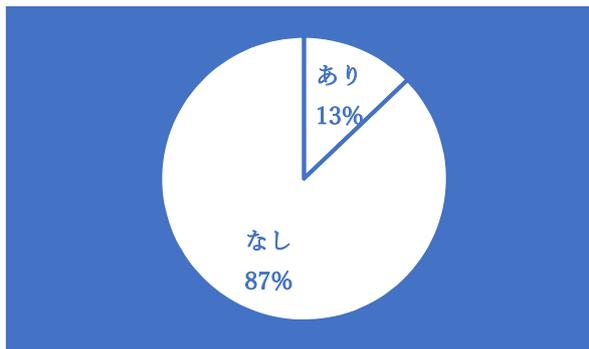


図 4 これまでのプログラミング経験について (2017 年度)

図 4 は 2017 年度受講者のプログラミング経験の割合を表したものである。2017 年度の受講者のうち、プログラミング経験者が使ったことのある言語は C、C++、JavaScript、Basic、PHP などであった。アンケートには書かれていなかったが、Python を使って独自に課題を進める学生もいた。

受講理由には「パソコンを活用するための知識を身につけたい」、「プログラミングに興味がある」、「プログラミングをしてみたかった」、「将来必要」、「役に立ちそう」など、2016 年度と同じ理由のものが多かった。

3 プログラミング入門の実践

3.1 使用言語について

授業で用いた言語は平山尚氏により作られたプログラミング言語「Sunaba」[6]である。この言語を選択した理由は次のようなものである。

- 日本語が使えるので初心者にはとっつきやすい。
- 限られた機能しかないので教える文法が少ない。
- 基本的な機能のみしか持たないため、プログラムを基礎的な部分から構成することになり教育的である。
- 丁寧に書かれた文献[6]が存在する。
- 導入（インストール）が簡単で Windows にも Mac にも対応している。
- 文法ミスに対するエラーが日本語なのでわかりやすい。
- コンパイルの作業が必要なく、ドラッグアンドドロップで即時実行できる。

3.2 授業内容について

授業ではプログラミング言語に Sunaba を使い、文献[6]に載っているような簡易テトリスの作成を最終目標とした。簡易テトリスは、枠の中でテトリミノ（4つの正方形がつながったもの）が上から落ちてきて、そのテトリミノをキーボードを使って左右移動や回転させることができるものである。もちろん、四角形が一行に揃うとその一行は消され、上に載っている四角形が下に落ちてくるので簡易とはいえテトリスらしさは楽しめるものとなっている。図 5 に簡易テトリスの一画面を示す。

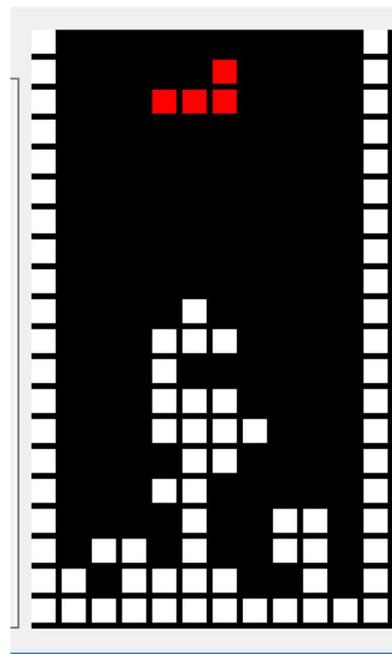


図 5 簡易テトリス

また、Sunaba を使って簡易テトリス作成に至る全 15 回の授業の具体的な内容（2017 年度分）を表 2 にまとめる。

表 2 授業内容（2017 年度）

第 1 回	Sunaba インストール、動作確認
第 2 回	点を打つ、四角形を書く
第 3 回	繰り返し構文
第 4 回	部分プログラム、枠の作成
第 5 回	部分プログラム
第 6 回	四角形を落下させる

第 7 回	キーボードを使って点を動かす
第 8 回	条件実行文
第 9 回	四角形を積もらせる
第 10 回	一列に並んだ列を消す
第 11 回	落下する四角形を 2 つにする
第 12 回	2 つの四角形を回転させる
第 13 回	落下する四角形を 3 つに拡張する
第 14 回	落下する四角形を 4 つに拡張する
第 15 回	まとめ

3.3 授業方法

実際の授業では学習者も使うことができる大学の教務情報システム(KULAS)の機能を活用し、資料配布、課題提出、出席確認、アンケートなどを実施した。

毎回講義用スライドを授業数日前に KULAS に載せることとし、スライドには前回の復習と次に学ぶ新しい内容を含めた。これは授業の課題をなるべく授業中に終わらせるよう事前学習（復習と予習）を促すために行ったもので、反転学習となることを意識している。スライドのおおまかな流れは以下の通りである。

- アンケートに書かれた質問に対する回答
- 前回の復習
- 前回の課題解答例
- 今回の新しい内容
- 今回の課題

課題は主にその回に学んだ内容を定着させるものとし、毎回課すようにした。課題提出物としてテキストファイルのプログラムそのものを提出させる回もあれば、課題を画面表示しそれを画像に保存して送付させる回もある。例えばキーボードを使って点を動かす回では、点を書いては消すの繰り返しのプログラムを作ることによって点を動かすことにした。そのときには、作ったプログラムを利用して図 6 のような一筆書きができるようになることを課題とした。ほんの少しの気付きによりこなせる課題であるが、見た目がガラッと変わるため、学習者の反応がよい課題であった。

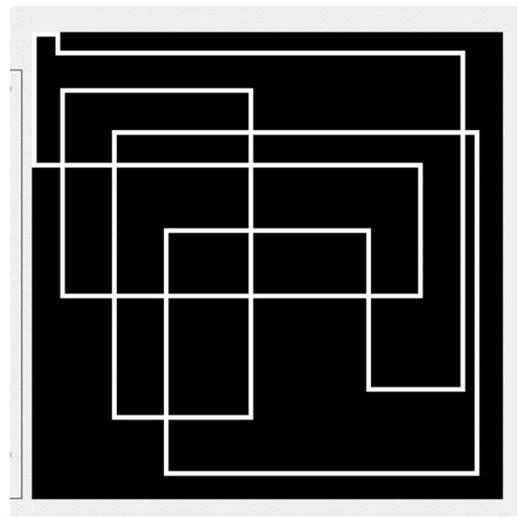


図 6 課題例（一筆書き）

また、授業中に多くの人が躓いた内容として繰り返し構文や部分プログラムなどが挙げられる。これらの項目については授業で繰り返し行い、使い方に慣れるためにそれを使う課題を何度も課すことにした。図 7 は繰り返し構文の課題の一例で、四角形を使って市松模様を描かせるものである。繰り返しの入れ子構造を使えるようになることを意図した課題である。

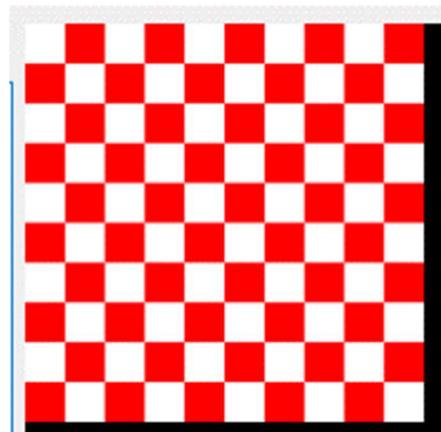


図 7 繰り返し構文を使った課題（四角形で市松模様）

また KULAS を使って、出席代わりにアンケートを毎回行い、どこで躓いているのか、どこがわかりにくいのかを把握し、次回以降にフィードバックするようにした。

授業の半分、あるいは 3 分の 1 ほどがスライド説明で、残りは課題作成のための時間にあてるこ

ととした。本講義には TA など補助者がいないため、課題作成の時間を多く取って教室を回り、個別対応するよう心掛けた。共通教育での授業であるため、受講生の所属がバラバラとなり個々で課題をこなしている学習者が多く、また人数が多いと個別対応にも限界があるため、2017年度は躓きやすい項目のところでグループ学習を取り入れることにした。しかしながら、出席を厳しく取っているわけではないため、欠席者が多い回はグループを作ることが難しく、グループ学習は数回行ったのみであった。一部の学習者からはグループ学習は有効との声もあったが、グループ学習そのものをしたくないという声も多く、今後の検討課題の一つとなった。

授業にまじめに出ていた学生は 15 回の授業で簡易テトリスを完成させることができ、最終課題に取り組んだ。成績に大きく響く最終課題は、Sunaba をダウンロード[7]すると付属している豪華版テトリスの一部機能を自分が作った簡易テトリスに取り入れることとした。これは、他人の書いたプログラムを読む能力、読み取ったものを自分のプログラムに組み込む能力が養われているかどうかをみるためである。

3.4 授業後アンケート

15 回の授業終了後、授業全体に関するアンケートを実施した。

プログラミング言語として Sunaba を使用したことについては次のような意見があげられた。

- 英語ではなく日本語でプログラムの指示ができるということがとても新鮮に感じた。
- プログラム実行の際にどこか間違っていたらどこがどう間違っていると示してくれるのがありがたかった。
- 受講前は「プログラミング＝複雑な数式や文字の羅列」というイメージがあったが、Sunaba は使用者にもわかりやすいプログラムを組むことができ、初歩を学ぶには最適。
- Sunaba についての情報がインターネット上にあまりなく、うまくできなかったときに調べようと思っても調べられなくて苦労した。

- プログラミングにはそれ特有の難解なワードが書き連ねられている、というイメージを持っていたが、Sunaba は日本語でプログラミングができ、非常にわかりやすかった。
- 自分の書いたプログラムが、すぐにビジュアルに反映されるので、プログラムを作っているという実感があって良いと思った。

最終目標をテトリス作成としたことについてのアンケート結果は図 8 のようになった。

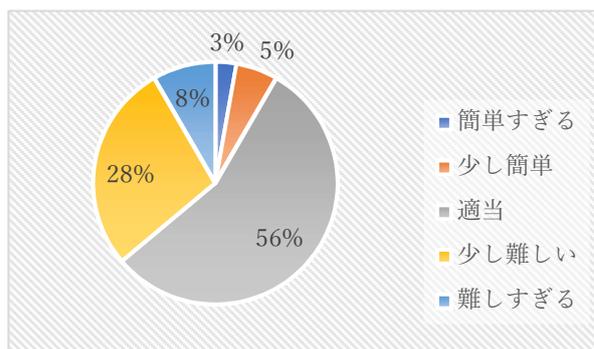


図 8 授業でテトリスを作成したことについて (2017年度)

また、本講義を受ける前と比べてプログラミングの考え方が身についたと思いますか？との問いには図 9 のような回答が得られた。

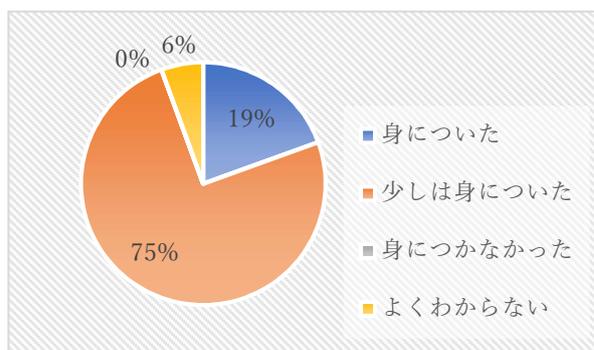


図 9 プログラミングの考え方が身についたか？ (2017年度)

4 おわりに

本稿では、本学で行った共通教育におけるプログラミング入門講義の実践内容を報告した。

“はじめに”の項で述べたように、本学ではプログラミングを学んでみたいという要望をもつ学生はそれなりに存在しており、その要望を満たす入り口となるようなものになればと考え、本講義

を開講した。共通教育でのプログラミング入門なので対象者はプログラミング未経験者とし、プログラミング言語には入門用言語の Sunaba を使用した。Sunaba はプログラミングに慣れているものには日本語と半角でのタイピングがわずらく、限られた機能しかないため使いやすいとはいえないが、初心者にはわかりやすい言語であった。また実行時のエラーがわかりやすいため、プログラミング未経験者でも自分で問題を解決しやすいようであった。一方教える側からすると、機能が限られている分同じことをするにも工程数が多くなるのでまどろっこしく感じられるが、逆に考えると順にステップを踏むことになるため、ある意味教えやすく感じられた。第 1 学期に開講したこともあって、受講生にはすでに情報処理を学んだ 2 年生以上と情報処理を学んでいる最中の 1 年生が混在していた。そのため、PC 操作の速さには個人差があったものの、Sunaba を使う分には情報処理を学んでいるかどうかはあまり関係なくすぐに慣れたようであった。

授業法としては、1 年目より反転学習を意識して教材を開講前に KULAS に載せ、予習の定着を図るようにした。2 年目はそれに加え、理解しにくい項目（特に繰り返し）に時間をかけるようにした。また課題作成が中心となるため、グループ学習なども取り入れた。

全 15 回の授業を使ってテトリス作成という一つの目標を達成するには、毎回のプログラミング経験の積み重ねが必要となるため、開講前は学習者が最後までついてきてくれるのか不安であった。しかしながら、テトリスは目標が明確であり、徐々にテトリスらしい動きが出てくるためプログラミングをしているという実感がわきやすく、やる気が長く続くようであった。実際に学生からは「初めはテトリスができるとは到底思えなかったが、授業が終わると実際にできていた。」「難しいけど、できたときの達成感はずごかった。」「テトリス楽しかったです。目標課題をテトリスにして正解だったと思います。」といった声が聴かれた。

Sunaba を使ったプログラミングは上級言語を使うための技術的な架け橋にはおそろくないが、一つのゲームを作るという作業を通してプログラミングの考え方的一端には触れられるものとなったのではないと思われる。次年度以降はプログラム作成の道筋をよりすっきりと示すことによりテトリス作成を少ない回数で終わらせ、Python などの上級言語を使ったプログラミングにも触れられるよう工夫するつもりである。

今後小学生からのプログラミング教育の必修化がなされると、大学入学者は小中高においてプログラミングを学んでくることになり、大学においてもそれに合わせたプログラミング教育を実施しなければならないことが容易に予想される。これは共通教育においても入門的な講義だけでなく、より実用的なプログラミング教育を行うことができる可能性があることを意味している。この先、入学者が小中高の各段階でどのようなことを学んでくるのかを注視し、今後の共通教育におけるプログラミング教育の内容を検討していくべきであろう。

参考文献

- [1] 日本再興戦略 2016、http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/2016_zentaihombun.pdf、(2017 年 9 月 30 日閲覧)
- [2] 未来投資戦略 2017、<http://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/miraitousi2017.pdf>、(2017 年 9 月 30 日閲覧)
- [3] 若年層に対するプログラミング教育の普及推進
http://www.soumu.go.jp/main_sosiki/joho_tsusin/kyouiku_joho-ka/jakunensou.html、(2017 年 9 月 30 日閲覧)
- [4] 岡部成玄、“一般情報教育の全国実態調査”、情報処理学会誌、Vol.55 No.12, 1400-1403 (2014); Vol.56 No.1, 94097 (2015)
- [5] 小学校学習指導要領、http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2017/05/12/1384661_4_2.pdf、(2017 年 9 月 30 日閲覧)
- [6] 平山尚、「プログラムはこうして作られる」、秀和システム、2013 年
- [7] Sunaba, <https://hirasho.github.io/Sunaba/>、(2017 年 9 月 30 日閲覧)