

E-03

文系学部低学年を対象とした情報リテラシー教育でのプログラミング Programming in information literacy education for freshmen of liberal arts course

宮本 友介[†]
Yusuke Miyamoto

中西 通雄[‡]
Michio Nakanishi

安留 誠吾[‡]
Seigo Yasutome

1. はじめに

高等教育における「2006 年度問題」では、2003 年度に実施された高等学校学習指導要領の改訂にしたがって新たな教育を受けた学生が大学等に入学することにより、学習内容の多様化に対応することが求められた。一般情報処理教育では、高等学校において教科「情報」が必修化されることにより、従来おこなわれていた項目が前倒しで学習されていることになることから、教育課程の内容についての見直しが議論された。

われわれは、この「2006 年問題」への対応として、文系学部低学年を対象とした一般情報処理教育科目の中で、約 10 年にわたって初学者向けプログラミング環境 PEN を用いたプログラミング演習おこなってきた。また、今年度は演習に先立って Scratch を用いた導入を取り入れた。本稿では、その経緯および効果について報告する。

2. プログラミング演習

2.1 導入の経緯

従来、一般情報（処理）教育としてはプログラミングがあまり取り上げられてこなかったが、その理由としては「専門課程との接続性が低いと思われる」という点と「プログラミングの習得には一定の時間が必要である」という点が挙げられる。

前者については特に文系学部では深刻であるが、高度情報化が進む中で、人材育成面での立ち遅れの原因となっていると考えられるだろう。情報処理学会情報処理教育委員会は 2005 年に「日本の情報教育・情報処理教育に関する提言 2005」を示し、その中で「多くの国民による、実践力を伴う『情報処理の理解』が必要」と指摘している。ここで、「情報処理の理解」とは「コンピュータの本質は『手順的な自動処理』であることを、体感的かつ具体的に理解している」ことであり、単に知識としての理解だけではなく、実際に経験することの重要性が強調されている。そして、「手順的な自動処理」を構築する過程として、直ちに挙げられるのがプログラミングである。

一方で、後者の習得にかかる時間については、近年さまざまな初学者向けプログラミング環境が発表されている。これらを活用することで、実用的なプログラミングの習得にまでは至らないとしても、上記提言で挙げられた「手順的な自動処理」の体感的・具体的な理解を促すことが可能であると考えられる。

こうしたことから、われわれは文系学部の低学年を対象とした一般情報教育の中でプログラミング演習を取り入れることにした。

2.2 PEN によるプログラミング演習

PEN (Programming Environment for Novices)^[2]は、初学者向けプログラミング環境であり、大学入試センター試験用手順記述標準言語 (DNCL) を拡張した分かりやすいプログラミング言語を備え、日本語を使った制御構造の記述が可能である。また、プログラム入力支援機能によって、直感的に操作でき、構文エラーの発生を抑えられる。実行トレース機能や簡単な図形描画機能を持っており、比較的容易にプログラミングを学ぶことができる。さらに、PEN は Java プラットフォームで実装されたアプリケーションソフトウェアであり、無償で提供されているため、受講生が容易に自律学習できる点にも配慮した。

2015 年度までの授業では、学期後半の 90 分×4 回の時間を用いて、基本的な制御構造から順に解説する標準的な教授法と、まず図形描画の課題を与え、それに必要なプログラミングの要素を解説する課題駆動型の教授法の比較検討をおこなった。各回の授業アンケート結果によると、課題駆動型の教授法では、初期の動機づけを高めるのに有効であるが、次第に別の課題に取り組み意欲は低下する傾向が見られ、課題設定には工夫が必要である。

2.3 Scratch による方向づけ

2016 年度は、PEN による演習に先立って、30 分程度の時間で Scratch^[2] ver. 2.0 を用いた方向づけを取り入れた。

Scratch は MIT メディアラボによって開発された、主に子どもを対象としたプログラミング言語学習環境である PEN と同様、構文的な知識がなくとも直感的に操作可能であることが重視されており、あらかじめ用意された「ブロック」を組み合わせることでプログラミングをおこなう。こうした仕組みにより、特に学習の初期段階においては PEN よりも Scratch の方が短時間で「手順的な自動処理」を体感できると思われる。そこで、まずは Scratch でプログラミングの全体像を概観しておくことで、後の演習を理解しやすくなることを期待した。

実際に作成したプログラムは、スプライトを動かすだけの単純なものだが、たとえば図 1 に示すような「10 歩ずつ

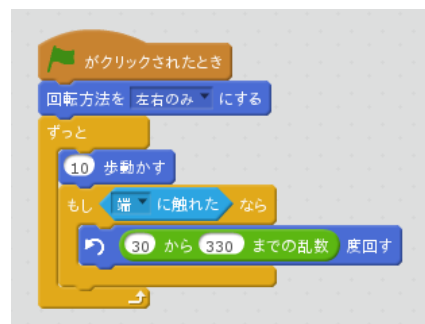


図 1: Scratch のプログラム例

[†] 大阪大学, Osaka University

[‡] 大阪工業大学, Osaka Institute of Technology

進んで、画面の端まできたら方向を変える」といった「条件分岐」と「繰り返し」の組み合わせが必要となるものである。また、「図形描画」や「変数」の使用についても触れている。

2.4 従来との比較

Scratch による方向づけの効果を検証するため、PEN のみで演習をおこなった 2015 年度との比較をおこなった。まず、「変数」「条件分岐」「繰り返し」といったプログラミングの構成要素について、どの程度理解できたかの自己評価を確認した。(図 2~4)

その結果、期待とは逆に、2015 年度の方がプログラミング構成要素に対する理解度が高いことが分かった。各項目の平均値(標準偏差)は、変数: 3.27 (1.41) > 2.66 (1.18)***, 条件分岐: 3.58 (1.28) > 142 (3.12)***, 繰り返し 3.36 (1.37) > 3.08 (1.10)*** であり、統計的に有意な差がみられた。

一方、「プログラミングは楽しいですか?」という質問項目には、2015 年度と 2016 年度では有意な差はみられなかった。

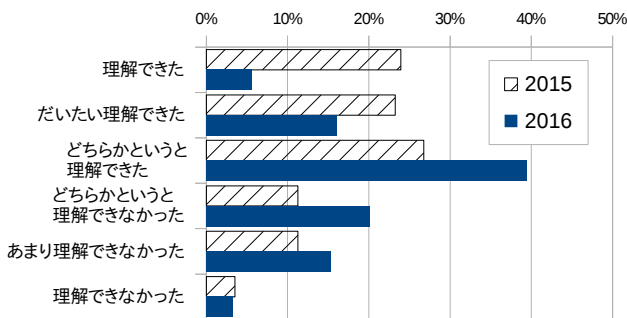


図 2: 変数について

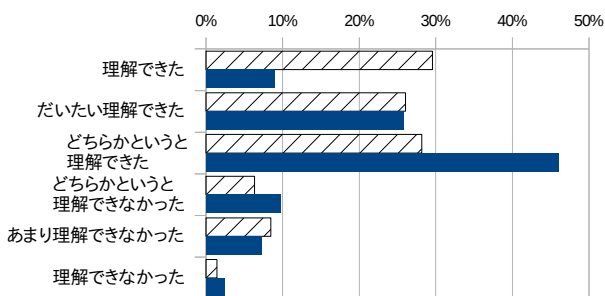


図 3: 条件分岐について

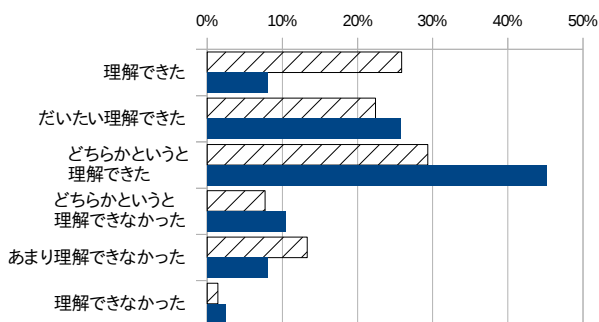


図 4: 繰り返しについて

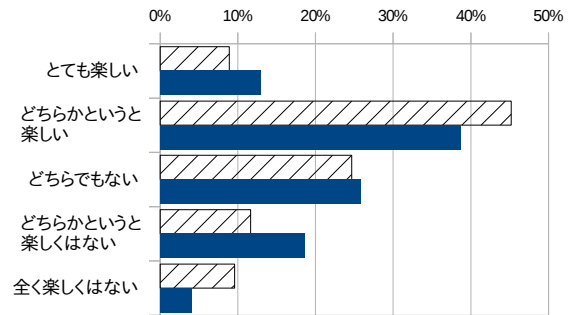


図 5: プログラミングは楽しい?

3. 考察

受講生を対象に高等学校までにプログラミングを(独学も含めて)学習した経験の有無についてアンケートで調査したところ、図 6 のような結果が得られた。年度によって前後するが、次第に増加する傾向がある。2016 年度では、30% を超える受講生が事前にプログラミングの経験を有していた。プログラミング経験の有無により、Scratch と PEN に対する態度に乖離があるように思われる。アンケートの自由記述によると、初めてプログラミングを体験した受講生にとっては、Scratch による方向づけは「とっつきやすくよい」と肯定的に捉えられているが、PEN での演習では「難しい、ついていけない」と否定的な態度になってしまいう傾向があるようである。一方、プログラミング経験のある受講生では、Scratch を「もの足りない、(ブロックを組み合わせる)操作が面倒だ」と否定的に捉える傾向があり、PEN については「面白い」と評価する意見や、「そもそも日本語はプログラミングに向かない」「もっと実用的なプログラミングをしたい」といった意見もみられる。

Scratch によるプログラミングの理解度は比較的高いにもかかわらず、演習全体としては理解度が低くなった理由としては、初学者にとっては構成要素の一般化が不十分であり、異なるプログラミング環境での適用が困難であるからと考えられる。いかにしてプログラミング環境の一般化した理解を促すかについては、今後の検討の対象としたい。

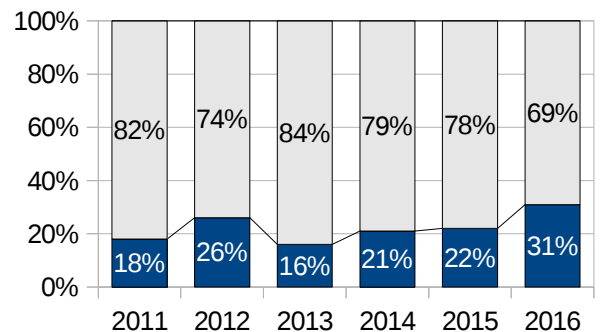


図 6: プログラミング経験の有無

参考サイト

- [1] <http://www.media.osaka-cu.ac.jp/PEN/>
- [2] <https://scratch.mit.edu/>