

初学者を対象とした授業用プログラミング学習環境の構築

布施 泉^{*1}, 中原 敬広^{*2}, 岡部 成玄^{*1}

^{*1} 北海道大学 ^{*2} 合同会社三玄舎

Learning Environment for Programming in Beginner's Class

Izumi FUSE^{*1}, Takahiro NAKAHARA^{*2}, Shigeto OKABE^{*1}

^{*1} Hokkaido University ^{*2} Sangensha LLC.

The purpose of this research is to investigate the learning environment of programming for beginners in a class by evaluating the environment through construction of its prototype and concrete practices. We have developed a learning environment of programming as a Moodle plug-in so that learning history of learners can be managed. We have implemented a learning environment which allows learners to browse learning materials, create a program and execute it on PC. We have used the learning environment for beginners in a class of 2016. According to the self-evaluation of the learner, the degree of satisfaction for the lesson is generally good on an average, although the lesson is judged to be harder on an average. This shows that the implemented learning environment of programming is effective for beginners.

キーワード: プログラミング, 学習環境, 授業, 大学, 初学者

1. はじめに

本研究の目的は、初学者が授業で用いるためのプログラミング学習環境を検討し、プロトタイプのプロγραμμαング学習環境の構築とそれを用いた具体的な実践による当該学習環境の有効性の評価を行うことにある。

初学者に対するプログラミング教育は、初等中等教育段階から高等教育に至るまで、様々な教育実践がなされている。例えば岡本らは、プログラミング学習における模倣の重要性に着目し、初学者のプログラムの模倣におけるつまずきについて、認知的負荷理論を用いた類型化を行い、非本質的な認知的負荷が要因となるつまずきを、教材の工夫により低減できることが示した^[1]。関谷らは、初学者のトレーシングにおいて誤りやすい解釈に基づく誤った出力を得る仕組みを作成することで、初学者の誤答パターンの分析を行った^[2]。兼宗らは、オンラインで利用可能なプログラミング学習環境を提案し、複数のプログラミング言語に対応可能な統合環境を提案している^[3]。

どのようなプログラミング言語を、どのような学習環境でどのように用いるのが初学者のプログラマ

的思考の向上につながるかは、発達段階に応じ、今後とも実践を進めていく必要がある。現時点では確固たる方法が確立しているわけではなく、研究は発展途上であると考えられる。本研究では、このような背景を踏まえ、大学の一般教育としての授業を前提に、初学者のためのプログラミング学習環境のあり方について、検討を進めることとする。本研究では、実際の授業形態を踏まえ、学習者の認知的負荷の低減を図り、学習者がプログラムを思考する学習活動を活性化するために必要とされる学習環境のあり方について検討を進める。

2. プログラミング学習環境の要件

本論文では、以下の学習者と学習環境を想定する。

1. 主に大学生を対象とする。
2. 初学者を対象とする。
3. 授業での利用を前提とするが、授業以外でも同様の環境で利用可能で、学習者管理ができること。

大学生が対象であり、プログラミング言語は汎用性があるものを検討する。本論文では Ruby を用いた実

践を示す。授業での利用を前提としており、履修者への授業資料の提示が簡便な環境が必要である。また、授業であり、成績評価を行うことが必要となるため、学習者の授業時間内外での利用状況を把握できると進捗管理などを行いやすい。

原則として、初学者を対象とした環境の構築に特化し、熟達者になった際には、各自で設定する独自環境へ移行することも念頭として設計する。そのため、当該環境特有の操作は極力排除することとする。

初学者が対象のため、プログラミング行為そのものに注力できる環境を目指し、プログラムの参照、入力、実行に際し、画面遷移や別アプリケーションの利用など、プログラムを思考する以外の認知的負荷が極力かからないような学習環境を検討する。また、プログラミングを理解し、自分のものとして活用するには時間を要するため、授業時と同等な学習環境を、いつでもどこでも提供できる仕組みが望ましい。さらには、学習間で協調的にプログラムを利活用し、グループ活動として、個々のプログラミングを組み合わせ、ある用途を達成するようなプログラムの作成を行うこともできる学習環境を検討する。

上記の要件から、本研究では、Moodle プラグインとして、プログラミング学習環境を構築することとした。Moodle であれば、個人がアカウントを持つことで、特定の Moodle コース上で、プログラミングの学習活動を行うことができる。学習の成果として生成されるプログラムや閲覧資料等は、当該アカウントがアクセスできるディレクトリ内に保持されるため、ブラウザがあれば、いつでもどこでも、学習を継続して行うことができる。さらに教師権限で、学習者の課題の進捗管理も原理的に可能となることが期待される。

3. プログラミング学習環境の構築

前章における要件から、初心者がプログラミングに注力できる環境を目指し、次のような学習環境を構築した。

図 1 に、開発したプログラミング学習環境の初期画面を示す。タブとして、「フォルダ」、「ライブラリ」、「ギャラリー」、「管理」（教師権限のみ）、（Moodle の）「コースへ戻る」、の 5 つを有し、デフォルトでは、「フ

ォルダ」タブ画面を表示することとした。

初学者である学習者がプログラミングそのものに注力できるようにするための工夫の一つとして、Moodle 上でプログラミング学習環境に遷移した時点で、Moodle で通常はデフォルト表示されるメニュー等、プログラミングに関係しない表示は全てなくしている。「コースへ戻る」タブの押下で、通常の Moodle コースへ戻る。

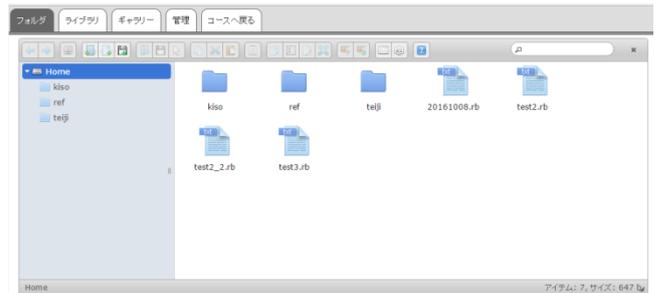


図 1 プログラミング学習環境画面（フォルダタブ）

3.1 「フォルダ」タブ

プログラムを編集、実行するための基本機能を有するタブであり、図 1 が一例である。フォルダ内にフォルダを生成することもできる。本フォルダの機能は、修正 BSD ライセンスの下に配布されているオープンソース elFinder^[4]を基盤として開発を行った。本環境では、elFinder の設定を編集し、右クリックメニューに「公開」「実行」を追加した。

フォルダ内のプログラムファイルを開いた際のエディタ画面例を図 2 に示す。フォルダ内のファイルをダブルクリックするか、マウス右ボタンによるメニュー「開く」の選択により、図 2 の画面に遷移する。複数のファイルを同時に開くことも可能である。

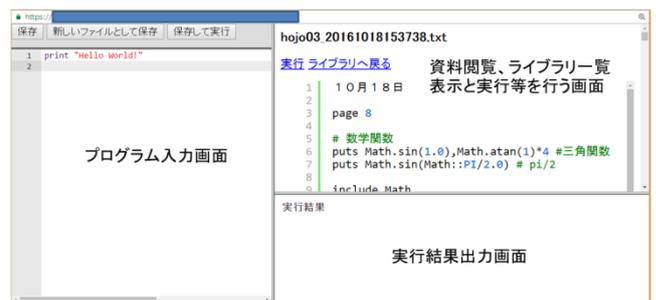


図 2 プログラムの入力・資料閲覧・実行結果画面

エディタ画面は3分割されており、左側にプログラムの入力画面がある。画面の上部に、「保存」、「新しいファイルとして保存」、「保存して実行」の3つのボタンを配置している。

一斉授業では、学習者は、教師が説明しているプログラムや資料等を閲覧しながら、プログラムを入力あるいは編集し、実行することが多いと考える。プログラムの内容を考え、入力し、実行結果を確認し、内容を評価し改善していくことは、学習者のプログラミングに関わる重要な思考活動である。可能な限り、その思考活動に注力できる環境が必要である。図2では、学習者がサンプルプログラムを参照し（あるいは独自に入力し）、実行するまでの手順を、ブラウザ上の一画面で完結できるようにした。授業では、教師の説明に合わせて資料を閲覧し、当該資料をそのまま実行したり、資料内容を一部選択したものを各自のプログラム編集画面上に複製し、修正して実行することも、画面遷移なくできる。このように、プログラムの編集や実行を、画面を3分割することにより、単一画面で完結できるように実装している。なお、画面フレームの境界は可変である。3分割した各フレームの主たる役割を以下にまとめる。

- ・ライブラリフレーム(図2の右上):本システムでは、サンプルプログラムや資料等、教師を含む参加者から提供された資料やプログラムをライブラリとして取り込むことができる。図2では、ライブラリとして取り込んだ資料の内容を右上の分割画面に表示している。この右上の画面上に青字で「実行」「ライブラリに戻る」というリンクがあるが、「実行」リンクをクリックすることでライブラリ画面にあるプログラムを実行することができる。「ライブラリに戻る」ではライブラリの一覧表示と選択表示を切り替えることができる。これらのライブラリは、後述する「ギャラリー」に公開されているファイルの中で、教師により許可されたファイルを各自で取りこむことで設定する。取りこんだライブラリは、各自が使いやすいように、フォルダに分けたり、不要となったものを削除したりすることもできる。
- ・エディタフレーム(図2の画面左):プログラムの入力・編集を行うフレームである。エディタフレームは、修正BSDライセンスの下で配布されているオー

ンソースのコードエディタ Ace^[5]を利用した。編集中のコードは、フレーム上部の「保存して実行」ボタンをクリックすることにより、即時実行され、実行結果は、右下の実行結果フレームに表示される。

- ・実行結果フレーム(図2の画面右下):エディタ上のプログラムを「保存して実行する」ボタンを押下することで、実行結果を表示する。Ruby(.rb), HTML(.html/.htm), テキストファイル(.txt), SVGファイル(.svg)の形式に対応し、拡張子を判別して実行する。また、本環境で学習し、熟達した学習者が、Windows環境での学習環境を整えた場合との互換性を考慮し、Rubyの関数や命令に関して独自の処理を行うことができるラップ機能を有したClassを準備した。

3.2 「ライブラリ」タブと「ギャラリー」タブ

プログラムや資料等のファイルは、各自でギャラリーに公開でき、学習者間の協調学習も可能とする設計にした。公開されたプログラム等は、「ギャラリー」タブ内に表示される。表示例を図3に示す。公開した日時をファイル名に付加し、ユニーク性を確保した。ギャラリー内のファイルは、教師が閲覧権限を設定できる。学習者が閲覧可能としたプログラムに対しては、図3の表示「アイコン」の下にある「操作メニュー」から、プログラムのプレビュー・実行(別ウインドウが立ち上がる)、利用者ライブラリへの追加、ギャラリーからの削除(公開した本人及び教師のみ可能)など、与えられた権限に応じた操作ができる。

「ライブラリに追加」から、図4の「登録」ボタンを押すことで、当該プログラムを利用者のライブラリとして登録でき、前述のライブラリフレームに表示できる。



図3 ギャラリータブにおけるファイル詳細表示



図4 「ギャラリー」タブからのライブラリ追加

4. 授業実践

北海道大学で第一著者を含む複数の教員が一般教育として開講している選択1単位のプログラミングの授業において、本システムを試行した。履修者は1年次の学生が多いが高学年も少数存在する。また、本授業は、北海道地区の国立大学での連携授業としても提供しており、遠隔の学習者も少人数(2016年度は1名)いる。

履修者はこれまでのプログラミング経験等を踏まえ、3つのグループに分かれて授業が進められる。授業内容は同等であるが、経験者と初学者では、授業進捗と方法が異なり、経験者グループでは、各自必要な資料を確認しながら各自で課題を進める形を取る。初学者のグループは、基本的に担当教員がサンプル等の説明や実行を行いながら、学習者に実行を指示する形の一斉型で授業を行っている(先に進みたい学習者は先取りでの学習も可能である)。本学習環境を用いたのは、この中の初学者向けの1グループである。

4.1 授業概要

本システムを利用したグループ(遠隔履修者を含む)の授業内容の概要と、各授業回において、本システムを使用した人数、並びにシステムにおける学習者全体での実行回数(一人当たりの平均実行回数を下段に付す)を表1に示す。遠隔の学習者を含むPC環境が異なる学習者に、同一環境をシステムとして提供できるメリットは大きいと考える。

表1 授業概要とシステム利用状況(2016年度)

週	日時	主たる内容	実行人数	実行回数
1	10/4	ガイダンス(大部屋)		
2	10/11	プログラムの入力・保存・実行・終了等の操作, 四則演算(課題1, 2)	24	1000 42/人
3	10/18	数学関数, 文字列, 基本入出力, 代入	22	1595 73/人
4	10/25	定数と変数(課題3), 比較, 条件分岐, 繰り返し	22	1075 49/人
5	11/1	繰り返し, ループ脱出(課題4, 5), 配列	20	1098 55/人
6	11/8	配列とハッシュ, ファイル入出力(課題6, 7)	21	654 31/人
7	11/15	課題7(魔法陣)の詳細, メソッド	20	623 31/人
8	11/22	メソッド(課題8), 再帰(課題9)	19	715 38/人
9	11/29	クラス, ビット演算(課題10)	19	699 37/人
10	12/6	正規表現(課題11)	19	805 42/人
11	12/13	応用課題:(例)円周率(モンテカルロ法ほか), ライフゲーム, スピングラス, 公開鍵暗号, アミノ酸の頻度分布, 文字の頻度分布, 金利計算等	19	518 27/人
12	12/20		17	521 31/人
13	1/10		13	299 23/人
14	1/17		13	266 20/人
15	1/24		13	266 20/人

学習者には、毎回の授業終了時に、作業記録を提出させている。作業記録には、毎回の授業で行った内容(箇条書き等自由記述)、質問事項(記述は任意)、授業進捗の満足度合(満足, まあ満足, 普通, 少し不満, 不満, から選択)、難易度(難しい, まあ難しい, 普通, 少し易しい, とても易しい, から選択)、その他のコメ

ントを記入することを求めている。

4.2 基礎課題

本授業では、表 2 に示す 11 の基礎課題の提出を求めている。学習者には、出席要件の他、これらの基礎課題をきちんと提出できていれば、単位は修得できる旨の説明をしている。

表 2 基礎課題一覧

基礎課題名	内容	必要知識（当該課題までの内容を含む）
課題 1	整数の演算	プログラム実行に至る一連操作
課題 2	浮動小数点数の演算（限界）	数値の表現
課題 3	変数を用いた計算	代入，出力
課題 4	繰り返しによる演算時間の確認	繰り返し
課題 5	平方根の逐次近似	ループ脱出
課題 6	HTML ファイルの出力	HTML 表示，ファイル出力
課題 7	魔法陣（行列の和を含む）	—
課題 8	メソッドにおける引数の設定	メソッドの作成，引数の対応
課題 9	パスカルの三角形（再帰）	再帰，書式付き出力（再確認）
課題 10	画像データの合成	ビット演算
課題 11	文字の頻度分布	正規表現

各課題の必要知識として、基礎課題 7 では、追加の学習事項はないものの、魔法陣のアルゴリズムの理解した上で、縦横のマス目の和の計算を独自に作成し、HTML 形式の表として表示する必要があり、難易度が高い（アルゴリズムについてはテキストで説明し、それに該当するサンプルプログラムも提示している）。例年、基礎課題 7 の前後で学習者の不満が高い傾向にあるが、2016 年度は特段の問題はなかった。

4.3 応用課題

第 11 回以降の授業では、基礎課題で理解した内容を踏まえた応用課題を学習者に取り組みさせた。この段階では、教師は TA とともに学習者の質問対応を行うのみで、一斉型授業は行っていない。応用課題では、SVG, Ruby/Tk などを用いて、グラフィカルな結果を描画する課題や文字の前後のつながりの頻度分布など基礎課題で扱った内容を少し高度にした課題など、多種多数の課題群を用意し、学習者が好きな課題を選択し無制限に解かせることとした。提出課題数の上限はない。また、基礎課題で使用していた Moodle 上の学習環境を用いなくても良いこととした。Moodle プラグインで Tk を利用し、ブラウザ上にグラフィカルな出力をレンダリングする機能を用意するのは今回の開発では困難であった。そこで、Tk を用いたプログラムを作成する場合には、学習者のローカル PC でプログラムの作成と実行を行う必要があること、ならびに処理手順についての説明を行い、ローカル PC での実施を促した。応用課題に取り組む段階では、実行に特段の困難はなく、学習者は好みに応じた課題選択を行った。

このような状況のため、学習者の課題選択内容に依存して、応用課題では Moodle 学習環境を用いない場合がある。表 1 で示す通り、11 回以降ではシステムの実行者数と実行回数が減少している。これは当該環境を“卒業”してプログラミングを実行していることに対応し、想定された動きと言える。

応用課題は、単にプログラムだけではなく、レポートの提出を課している。レポートには、当該応用課題における作業内容、課題概要、プログラムのポイント、結果や苦労したことや工夫したことなどを適宜まとめるように指示した。

図 5 は、応用課題として提出された円周率のモンテカルロ法での計算例である。上は Ruby/Tk で描画を行ったものであり、下は SVG で描画したものである。上はローカルマシンで、下は当該システムを用いて課題を実行している。

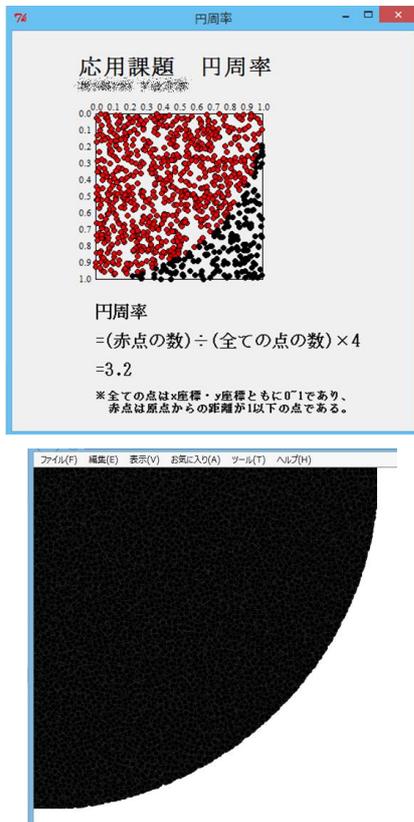


図 5 円周率をモンテカルロ法で求める応用課題の実行例 (Tk と SVG での使用例)

5. 結果と分析

本章では、開発した学習環境の有効性を評価するために、システムを用いていない 2015 年度と、システムを用いた 2016 年度の作業記録による学習者の自己評価結果を比較して示す。2015 年度、2016 年度ともに、学習者は初学者であり、同様のテキストで、同じ内容の授業を実施している。グループ人数は 2016 年度の方がやや多めであるが、概ね 20 名弱で同様である。2015 年度は、ローカル PC 上で、テキストエディタを用いてプログラムを入力、保存し、実行は Windows 上のコマンドウィンドウを開き、プログラムを指定して実行させた。

図 6 は、授業進捗に関する満足度 (5 段階評価) の結果である。満足度を 5、全く不満を 1 とした数値データとして、各回の平均値を求めたものである。2016 年度は本学習環境を用い、基礎課題に要する時間を 1 週間短縮することができた。基礎課題に取り組む時間が減少しているにも関わらず、基礎課題の取組期間における授業の満足度は 2015 年度のものに比して、概ね高いことが分かる。なお、2015 年度はシステム障害のために、第 4 週の作業記録は取れず、当該回のみは

比較すべきデータは欠損している。

各回の満足度について、ウェルチの方法による二群の平均値の差の検定を行ったところ、図 6 の点線で囲った第 7 週では、2016 年度の満足度が、2015 年度より有意に高い ($p < .01$) が示された。

難易度についても、難しいを 5、とても易しいを 1 と数値化した上で、同様の検定を行った。図 7 の点線で囲った第 9 週では、2016 年度の難易度は、2015 年度に比較して有意に低いことがわかった ($p < .05$)。この難易度のグラフは、2016 年度では、授業初期には少しずつ難易度が上がり、授業中盤で概ね一定 (やや難しい: 4 となる程度) するような推移をしており、望ましい状態であると考えられる。一方、2015 年度は、各回における学習者の感じる難易度の差が大きく、学習者が理解しやすい回と理解しにくい回が明確に分かれていることが示唆される。

2015 年度の第 7 週は HTML ファイルの出力と魔法陣を扱った回であり、HTML ファイルのタグ出力や魔法陣のアルゴリズムなど、内容の理解が比較的難しい課題を扱っている。2016 年度も当然同じ課題に取り組んでいるが、サンプルプログラムは複製でき、それを基に思考できるため、サンプルにおける入力ミスは発生せず、プログラムの内容の理解に時間をあてることができた。魔法陣の課題を取り扱った 2016 年度の第 7 回の満足度は、2015 年度と比して有意に高い。

応用課題では、各自が独力でプログラムを作成していくため、難易度はそれまでの基礎課題に比して高い。各自で取り組み、満足できる結果はすぐには出ないため、図 6 の通り、授業満足度も抑え気味に推移していることがわかる。この傾向は 2015 年度も同様である。

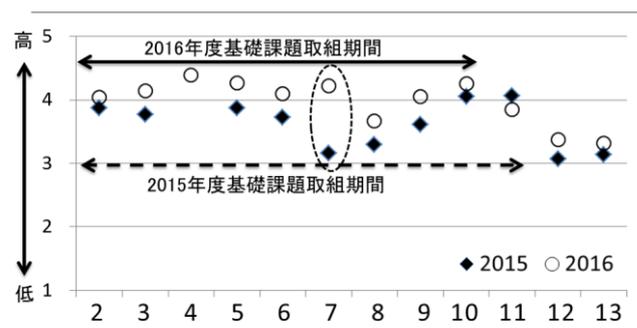


図 6 授業進捗の満足度についての学習者の自己評価。横軸は授業週数を示し、各授業時の満足度を数値データとして変換し、平均値を記した

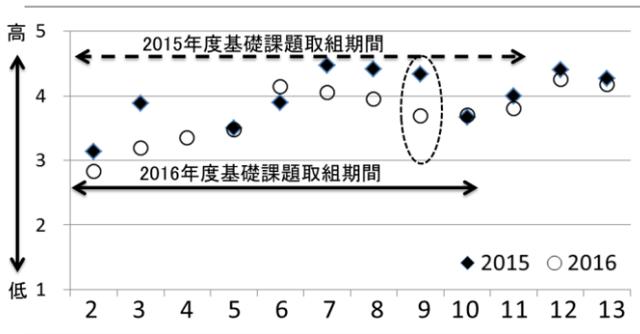


図 7 授業の難易度についての学習者の自己評価。横軸は授業週数を示し、各授業時の難易度を数値データとして変換し、平均値を記した

応用課題で提出されたレポートの内容を抜粋し、いくつか紹介する。

与えられたアミノ酸配列を読み込み、各アミノ酸の出現回数を数え、出現頻度順に並べ替えるヒストグラムを描画する課題に取り組んだ学生は、工夫点として、「プログラム簡略化のため、アミノ酸の種類を記した配列をプログラム内で作る作業を省き、aminosan_name.text というファイルで与えました。数える対象のファイルに関係なく、アミノ酸の種類は変わらないこと、また存在はするが数える対象の配列に含まれていないアミノ酸があるときはそのことが分かるようにしたかったことが理由です」と記述した。ライフゲームに取り組んだ学習者は、「Tk の仕様を理解するのに苦労した。また、初期段階では、描画の際にセルのオブジェクトを増やすプログラムとなっていたので、時間が経つにつれ、処理が遅くなっていた。これを改善するために、一度すべてのセルを消すというプロセスを増やした。」とプログラム作成の苦労と工夫を記述した。いずれも、課題の要求要件を自分で咀嚼し、自らの考えに沿ってプログラムを適切に作成していた。これは、本学習環境の利用により、基礎的な学習時間を短縮できたため、応用課題に比較的じっくりと取り組むことができたことと、基礎課題においてはプログラミングを思考する活動に注力できるようにしたことによる効果であると考えている。さらには、自宅でも同一環境で利用できる学習環境を用意できたことから、学習者によっては、授業時間外で毎週のようにならぬ3時間ほどシステムを利用していたものもいた。これらの結果から、初学者に対するプログラミング学

習環境として、本研究で開発したシステムは有効に機能していたと判断する。

6. 結論

本研究では、授業で利用することを前提とした大学初学者向けのプログラミング学習環境を構築し、その有効性を評価した。システムを用いていない 2015 年度と本システムを用いた 2016 年度の学習者の作業記録の分析から、本システムを利用することにより、学習者の授業進捗に関わる満足度が向上するとともに、難易度は回に関わらず、一定程度に抑え、全体として難易度を低減させる効果があることが確認された。これらの結果は、開発したプログラミング学習環境内で、教師の資料を確認し、プログラムを入力し、実行するという一連の操作が一画面で完結することによる認知的負荷の減少が理由の一つとして考えられる。

学習者のプログラム入力画面は、教師の資料からプログラムの一部を選択し複製でき、そのプログラムをそのまま実行することもできる。このようなコピーを許す実行環境には賛否があるかもしれないが、打ち込みミス回避することで、学習者がプログラミングそのものを思考する学習活動を増やすことはできる。実際、本システムを用いることで、2015 年度と比較し、1 コマ程度、基礎課題に要する期間を短くでき、その分を応用課題に取り組ませることができた。応用課題のレポートを確認する限り、基礎課題での教師プログラムをコピーすることによる弊害は見受けられず、総合的に学習者のプログラミングに関わる思考力向上に寄与できたと考えている。応用課題に至る際に、本学習環境を卒業する学習者もいたが、いずれも無理なく移行ができた。

このように、本研究で開発したシステムは、大学初学者向けのプログラミング学習環境として、有効であることが示されたと考えるため、今後は他のプログラミング言語への拡張や他の発達段階にある学習者への適用などを検討している。

謝辞

本研究の一部は、JSPS 科研費 15H02921、および 16H03074 の助成を受けた。

参 考 文 献

- (1) 岡本雅子: “模倣の重要性に着目した初学者向けプログラミング教育の研究”, 博士論文, 京都大学,
<http://repository.kulib.kyoto-u.ac.jp/dspace/bitstream/2433/188863/1/djohk00519.pdf> (2017年2月10日確認)
- (2) 関谷貴之, 山口和紀, 山本三雄: “初学者によるプログラミングトレーシングにおける誤答に関する分析”, 情報処理学会情報教育シンポジウム, 2012年8月, pp.113-120 (2012)
- (3) 兼宗進, 本多佑希, 林康平, 島袋舞子, 長慎也, 長島和平, 並木美太郎: “オンラインで利用可能なプログラミング学習環境の提案”, 日本情報科教育学会第9回全国大会, pp.89-pp. 90 (2016),
<http://kanemune.eplang.jp/data/jaeis1606jslesson.pdf> (2017年2月10日確認)
- (4) elFinder, <http://studio-42.github.io/elFinder/>, (2016年10月21日閲覧).
- (5) Ace, <https://ace.c9.io/>, (2016年10月21日閲覧).