

# 授業での利用を前提とした初学者用プログラミング学習環境の開発 —Ruby 言語の Moodle 環境での学習支援—

布施 泉<sup>1)</sup>, 中原敬広<sup>2)</sup>, 岡部 成玄<sup>1)</sup>

1) 北海道大学 2) 合同会社三玄舎

ifuse@iic.hokudai.ac.jp

## Development of Learning Environment for Programming in Beginner's Class - Support for Learning of Ruby Language in Moodle Environment -

Izumi Fuse<sup>1)</sup>, Takahiro Nakahara<sup>2)</sup>, Shigeto Okabe<sup>1)</sup>

1) Hokkaido Univ. 2) Sangensha LLC.

### 概要

本研究は、初学者対象の授業用プログラミング学習環境の開発を目的とする。学習者の学習履歴等を管理できるよう Moodle プラグインとして、プログラミング学習環境を構築した。サンプルプログラム等の授業資料を学習者に閲覧させ、プログラムを作成し、実行する一連の流れを一画面に統合して実装した。また、作成したプログラムを相互評価、相互コメントできるように設計した。2016 年度後期の授業で試行したが、学習者の自己評価によると、授業進捗への満足度は、2015 年度（システム未使用）と比して高く、難易度も適切と評価していることが明らかになった。初学者用学習環境として有効であることが示されたので、今後、学習者間のプログラム確認と評価等の協調学習への展開を予定している。

## 1 はじめに

情報化の中で、初等中等教育段階におけるコンピューティング教育は、近年、世界的に推進されている。英国では、小学校からのコンピューティング教育を 2014 年 9 月から必修で実施している。これらは、すべての子供たちに、コンピュータ科学の原理や概念の理解の他、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る。技術者を育成する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いる[2]。今後、初等中等教育での推進が強化されていくことは間違いないと考える。

現在、高等学校での必修科目である共通教科「情報」は、「情報の科学」か「社会と情報」のどちらかは学ぶことになっており、その比率は、2 : 8 と言われている[3]。前者の「情報の科学」では、プログラミングを行うことになっているものの、社会と情報では、そのような単元はなく、現時点では、学習者は、プログラミングを行わずに、大学に入学することが多い。北海道大学（以下、本学と記す）での調査でも、2016 年度はちょうど約 2 割がプログラミングを行った経験があると回答

している[4]。

学習者はプログラミングの経験がないものの、プログラミングへの興味が低いわけでは必ずしもない。第一著者は、本学の一般教育の枠組みの中で、プログラミングの授業を後期に半期開講している。毎回、履修希望者に履修したい理由を記載させているが、情報化の中で、プログラミングを試したい、行ってみたい、という希望が、毎年、少なからず見受けられる。ニーズはあるのである。プログラミング初学者は往々にして挫折をすることが多く、自力で学習しようとしたが、挫折したため履修しようと思ったといった回答もある。

本学は、必修の一般情報教育（前期）の中で、プログラミングを 1 コマ弱、時間を取って体験させている。時間に限りがあるため、プログラミング言語は Scratch を取り上げている。それを踏まえ、前述の後期に行うプログラミングの授業では、学習者には、汎用性があり、世界的に使われているプログラミング言語を初歩から学ばせることとしている。

このような背景から、本研究では、授業での利用を前提とした初学者用のプログラミング学習支援環境の開発を行い、当該環境の有効性を評価するとともに、開発システムに機能を活かした学習

手法について検討することを目的とする。初学者に対するプログラミングの学習支援手法は、様々な提案がなされている[5,6,7,8 など]。本研究で開発したシステムは、授業利用を前提とした初学者に特化した学習環境であり、LMS と連携した環境で、学習者間のプログラムの相互確認と評価が可能なフレームを持たせたことに特徴がある。

## 2 初学者用プログラミング学習環境

### 2.1 必要要件の検討

本論文で想定する学習者と学習環境に関する必須要件は以下の通りとした。

- ①初学者の大学生を対象とすること。
- ②授業での利用を前提とすること。教師が授業で利用するプログラム等の資料を、都度、学習者に閲覧させることが容易であること。また、学習者の進捗を容易に確認できること。
- ③学習者同士でプログラムの閲覧と評価ができるようにすること。ここで、学習者相互の閲覧と評価は教師の管理下でコントロールできること。
- ④当該学習環境は、自宅での授業外学習でも、同じ状態で授業時と同じ手順で利用できること。
- ⑤プログラムを書くエディタ画面と、実行画面を一つの画面で構成できるようにするとともに、スマートフォン等での編集と実行も可能とし、実行結果は、単体の画面でも表示できること。
- ⑥本環境は初学者を想定するものである。学習者が熟達した際には、当該環境を卒業し、学習者が設定した独自環境への移行が容易であること。

### 2.2 要件に応じたシステム環境の検討

前節の要件を踏まえたシステムを検討する。本研究では、初心者用の学習環境と割り切り、学習者が基本プログラムを作成、編集し、実行することに対し、そのプログラムの動作に注力できるユーザインターフェースを工夫することとした（要件⑤）。また、プログラミング言語は、熟達した場合にもそのまま利用可能な汎用性の高いものとし、オープンソースである Ruby を採用することとした（要件①、要件⑥）。

授業での利用を前提としているため、教師から学習者への資料の提示、および学習者のプログラム結果を教師が集約して確認できるような機能が必要である（要件②）。

初学者は、たくさんの誤りをする中で学ぶものである。様々なプログラムを確認し、自らの誤りを発見することも、同じ初学者の誤りを指摘して

学習者同士で修正を促すことも出来るような機能が必要であると考え。また、プログラムの見やすさを自分と他者のプログラムを比較することで学ぶことを想定し、学習者間の評価、コメントを可能とした設計とする（要件③）。また、一人でプログラミングの作業が可能な空間と、要件③による協調作業用の空間を、システム上で明確に分離することとした。

最後に、学習者は自らの学びを進めるために、自宅でも自宅外でも同等の環境で学習できることが望まれる。そのため、オンラインで利用可能な学習環境を要件とした（要件④）。授業内外、および自宅内外、どこでも自身の環境の中でプログラミングできるようにするために、認証を求めることとした。

上記の要件から、本研究では、このような学習環境を Moodle プラグインとして開発することとした。つまり、個人が ID を持ち、Moodle のコースにおけるメンバー内限定で、学習者間の協調作業を行うことができるものである。

## 3 開発システムの概要

### 3.1 Moodle プラグイン

図 1 は、Moodle のコースに入った直後の画面例である。コースの管理者である教師は、活動モジュールのインスタンスとして、プログラミング学習環境を設定する。

#### 誰でもプログラミング



図 1 学習環境例(Moodle コースの選択直後画面)

通常の Moodle であるため、当然ながら、開発したシステムとは別に Moodle の課題を本画面上で設定したり、資料を提示したり、といったことも可能である。ここで、プログラミング学習環境であるインスタンス（図 1 では「ここからプログラミング」と表記）のリンクを押下することで、図 2 のような学習環境画面に遷移する。図 2 は、教師用画面である。タブとして、フォルダ、ライ

ブラリ、ギャラリー、管理、コースへ戻る、の 5 つを有する。学生用画面には、管理タブが無く、残り 4 つのタブを有する。デフォルトでは、フォルダタブの画面を表示することとした。なお、本画面を始め、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、プログラミング学習環境に遷移した時点では、Moodle で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等は全てなくすこととして、開発を行った。「コースへ戻る」タブを押下することで、図 1 のような元の Moodle へ戻ることができる。

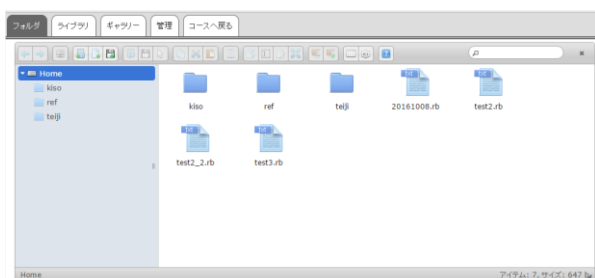


図 2 プログラミング学習環境画面（フォルダ）

### 3.2 「フォルダ」タブ

プログラムを編集、実行するための基本的な機能を有するタブである。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`[9] を基盤として開発を行った。`elFinder` は、`jQuery-UI` をベースとしているため、`jQuery`、`jQuery-UI` もフォルダタブ内では利用している。本環境の構築においては、`elFinder` の設定を編集し、右クリックメニューに「公開」「実行」を追加している。

図 3 は、図 2 の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の例である。フォルダ内のファイルをダブルクリックするか、マウスの右ボタンによるメニュー「開く」のどちらかにより、本画面に遷移する。複数のファイルを同時に開くことで、エディタ画面を複数立ち上げることができる。

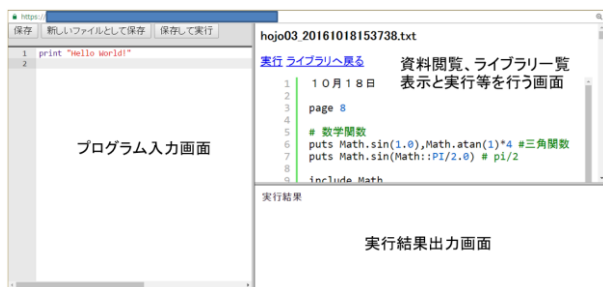


図 3 プログラムの編集・資料閲覧・実行画面

初学者は、授業時には、教師等のプログラムや資料を閲覧しながら、プログラムを編集し、実行することが多いと考えられる。このような学習を、画面を 3 分割することで、単一画面の中で完結できるように実装した。なお、3 分割の画面のフレームの境目は可変としている。以下、3 分割画面の各フレームの役割を記す。

- ・ライブラリフレーム（図 3 の画面右上）：その週に学習すべき内容、サンプルプログラム等、教師から提供された資料は、ライブラリとして取り込み、右上の分割画面に表示することができる。授業時には、これらを確認しながら、独自にプログラムを編集画面に入力しても良いし、そのままサンプルプログラムを実行して学習することもできる。図 3 では、特定のライブラリを表示した画面を示したが、ライブラリ一覧を表示することも可能である。ライブラリは、利用者が「ギャラリー」に公開されているプログラム等のファイルを、各自で取りこむ方式とした。
- ・エディタフレーム（図 3 の画面左）：プログラムを入力し、編集するフレームである。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プンソースのコードエディタ `Ace`[10] を利用した。`Ace` では、様々な言語に応じたシンタックスハイライトが用意されており、今回は `Ruby` 用に設定した。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、即時実行される。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 3 の画面右下）：エディタに記載したプログラムを実行した結果を表示するフレームである。この実行機能に対応しているファイル形式は、`Ruby(.rb)`、`HTML(.html/.htm)`、`テキストファイル(.txt)`、`SVG ファイル(.svg)` である。それぞれの拡張子により、判別を行って実行する。また、熟達した学習者が `Windows` 環境で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` を準備している。

ウェブブラウザ上での実行であるため、スマートフォン等からでも実行可能である。フォルダ内のファイルを右クリックし、メニュー「実行」を行った場合には、図 4 下図のように、ブラウザ上の別タブに実行結果が表示される。

また、図4のように、フォルダ内のファイルを選択し、右メニュー画面「公開」を押下した場合には、管理タブで教師が設定した設定項目の入力画面（デフォルトの設定項目は、カテゴリ、作品名、制作者、ライセンスである）が表示される。それらを入力し、「ギャラリー」への登録することで、コース参加者（もしくは教師のみ）に対し、プログラムを公開することが可能である。その公開範囲は、教師の管理下に置かれている。詳細は後述の「管理」タブで触れる。

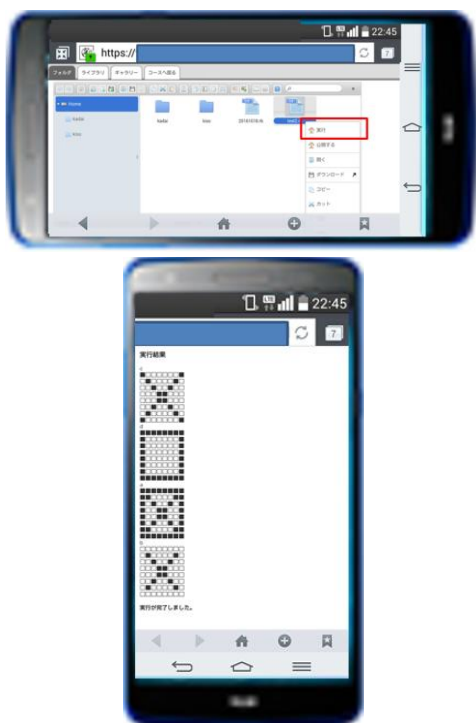


図4 スマートフォン上でのフォルダ表示画面からのファイル実行とプログラムの実行結果の例

### 3.3 「ライブラリ」タブと「ギャラリー」タブ

公開されたプログラム等は、「ギャラリー」タブ内で表示される。公開した日時をファイル名に付加し、同一ユーザが同名のファイルを公開した際にもユニーク性を確保した。学習者は、ギャラリー内のプログラムに対し、教師の設定した権限に応じ、特定のファイルのみ閲覧できる。閲覧可能なプログラムは、図5に表示されている「アイコン」の下にある「操作メニュー」から、権限に応じた操作ができる。具体的には、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、ギャラリーから削除（公開した本人及び教師のみ可能）、アップデート・非表示にする（公開した本人及び教師のみ可能）、非公開にする（管理者のみ可能）である。

「ライブラリに追加」を行うと、図6の通り、「ライブラリ」タブに遷移した上で、利用者としてのライブラリメモを合わせて登録することができる（ライブラリメモは入力しなくても良い）。学習者は、図6の「登録」ボタンを押すことで、当該プログラムを自身のライブラリとして、登録できる。

図5の画面右側の列には、学習者によるレビュー画面が表示されている。各プログラム等のファイルに対し、簡易評価ができる仕組みにしており、「評価する」、「コメントする」の2種のレスポンスが可能である。コメントは画面上にそのまま表記され、5段階で評価された評価点（★マークの数）は、評価した学習者での平均値が表示される。



図5 ギャラリータブにおけるファイル詳細表示



図6 「ギャラリー」タブでのライブラリ追加

### 3.4 「管理」タブ

管理タブは、教師のみがアクセスできるタブである。そのため、管理タブのみは、Moodleメニューをそのまま表示しており、各種設定をできるようにした。

本管理機能は、大きく、学習者がプログラムを公開する際のメタ情報の設定機能と、公開する際のカテゴリ設定機能に分けられる。メタ情報の設定は、デフォルトは、タイトル、作者、ライセンスの3種であり、コース管理者である教師が自由に追加することができる。また、公開時のカテゴリ設定は、図8のように、カテゴリの追加を可能とし



ており、カテゴリ名と公開設定（公開、登録者のみ、非公開の3種から選択）を記述、確認することで設定される。



図7 「管理」タブからの遷移画面

公開設定の「公開」は、コース内の参加者全てに閲覧権限（コードのプレビュー、実行）を付与するものである。教師の各週の資料等を提示することを想定している。また、学習者間で相互評価を行う場合にも用いる。公開設定の「登録者のみ」は、ファイルを登録した本人と教師のみが閲覧できる。教師の指示に従って、ファイルを教師に閲覧させる場合等に用いることを想定する。公開設定の「非公開」は、教師のみが閲覧できる。

これらの公開設定は、任意のタイミングで変更可能である。例えば、教師が授業時に用いる資料は当初は非公開としておき、授業で使う際に、「非公開」から「公開」にすることで、学習者はライブラリに取りこむことが可能である。また、学習者の提出課題は、当該カテゴリを当初「登録者のみ」とし、提出が完了した際には、「非公開」とすることを想定している。学習者間の相互評価においては、当初は「登録者のみ」とし、各自のプログラムを掲載し、その上で公開設定を「公開」とすることで、参加者に対し、一斉に公開することが可能である。



図8 管理タブにおけるカテゴリの設定画面

## 4 システムの試行評価

本研究で開発したシステムを学習者に試行させ、評価を行った。第一著者における2016年度後期の授業での学習者の主観評価を、2015年度後期の授業（授業内容と授業規模はほぼ同等：本システムは未使用）と比較することとする。

### 4.1 対象の授業

第一著者の授業において、開発システムを試行した。本学における一般教育として開講している1単位の選択授業で、プログラミングについて学ぶ授業であり、履修者は1年次の学生が多いが高学年も少数存在する。本授業では、履修者を、言語を問わないプログラミング経験の有無により、3種類のグループに分けて指導している。第一著者は、2015年度、2016年度とも初心者対象のグループを指導している。本学では、北海道内の国立大学での連携授業を行っており、当該科目は、連携授業として提供している[11]。少数ではあるが、遠隔地の学習者が存在する。

2015年度は、教育用のWindowsマシンにおけるローカル環境で授業を行い、2016年度は本研究で開発した学習環境にて授業を行った。特に連携大学の学習者に対しては、どこにいても、ネットワークにさえつながれば、本学の学習者と同等の学習環境を提供できるという意味でも、本学習環境の構築は望ましいものであると考えている。2015年度は、毎回、遠隔地の学習者に授業前日に5分程、学習環境の設定を求めることで授業を実施しており、2016年度はそれらの作業が軽減されている。

### 4.2 システム評価の方法と結果

初学者対象の第3グループに割り付けられた学習者は、コンピュータ教室のサイズから、各年度概ね20名前後である。学習者には、毎回の作業記録を記載させている。授業で行ったこと（必須）、質問（任意）、進捗度合（必須：満足、まあ満足、普通、少し不満、不満の5択から選択）、難易度（必須：難しい、まあ難しい、普通、少し易しい、とても易しい、の5択から選択）を記載させる。

2016年度の授業は現在進行中であり、本稿では、最初2回分の結果を示す。最初の2回の学習内容は、2015年度、2016年度とで同等である。初回は、Rubyのプログラムの書き方と実行方法の確認、整数と浮動小数点数を用いた四則演算を中心とし、課題提出の方法についての説明を行った。2回目

は、数学関数の記述、文字列の表記と操作、書式付き出力、代入等の扱いについて、プログラムを作成し実行させている。特に2回目は、様々な項目についての説明を行うため、学習者が混乱する可能性があり、学習者には、わからなくなった時点で、すぐに知らせるよう周知しているものの、質問は少なめである。

表1は、この最初の2回（導入）の授業における進捗度合と難易度についての評価結果を積算したものである。2015年度は16名延32回答分、2016年度は22-23名、延べ45回答分を集計している。

表1 学習者が感じる難易度と授業の進捗度合

難易度	2015	2016	進捗	2015	2016
難しい	16%	2%	満足	19%	33%
まあ難しい	31%	20%	まあ満足	50%	42%
普通	44%	62%	普通	22%	22%
少し易しい	9%	11%	少し不満	9%	2%
とても易しい	0%	4%	不満	0%	0%

学習者が感じる難易度は、「普通」を中心とし、「難しい」が最も少なく、次に「とても易しい」が少なくなっていることが望ましいと考える。初学者が導入時に「難しい」という印象を感じた場合には、プログラミング全般に対する悪印象を生じる原因となり得る。一方で、「とても易しい」との印象は、もっと進みたいという意欲の表れとも考えられるため、より理解を深める課題の提示が必要であると考えられる。

表1では、まず難易度において、一番注意を要する「難しい」と回答する学習者の割合が、2016年度で各段に減少していることが見て取れる。また、「とても易しい」とした学習者が2名いたが、ともに初回の授業での評価であり、同時に評価された満足度は高いことから、大きな問題はないものとする。普通が6割、まあ難しいが2割、少し易しい1割で、合計9割を超えており、理想的な難易度設定になっていることが示された。

次に表1の満足度を比較する。特に5段階で最高の満足度を示した学習者が2016年度は3割おり（2015年度は2割弱）、まあ満足、普通との評価を加えると98%に達するため、負担なくシステムを利用して授業に参加できていることが分かる。

これら満足度と進捗の度数分布を図9に示す。

2016年度では、満足かつ難易度が普通と評価する度数が高くなっていることが示されている。

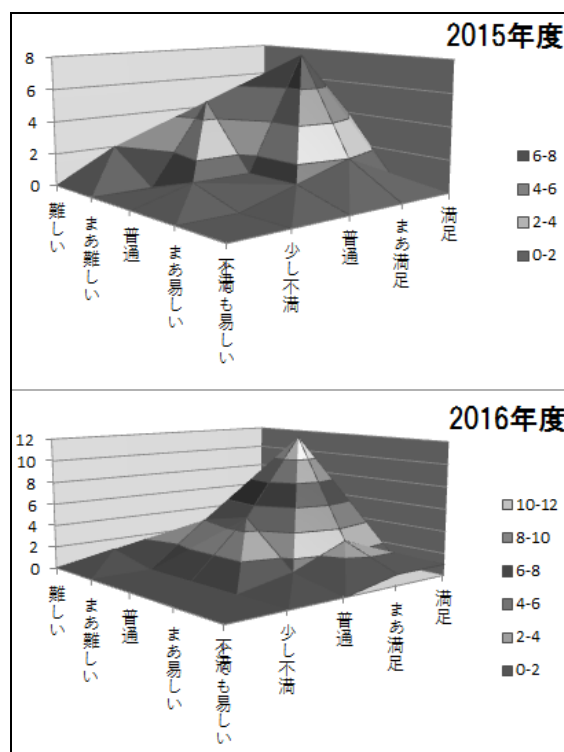


図9 満足度と進捗の度数分布の内訳

## 5 考察・授業での更なる活用例

### 5.1 評価結果に対する考察

前章での結果から、本システムは初学者の学習環境として、有効であることが示されたと考える。授業では、教師の指示する内容について、特段のトラブルの発生を示す学習者は殆どおらず、黙々と作業を行っている。集中してプログラムを行っているように観察されており、2章で求めた要件である同一画面での資料閲覧、プログラムの作成、実行の手順が有効に機能していることが理由の一つであると考えられる。2015年度の学習者の自由記述では、「スムーズにできるぎりぎりの難易度だったと思う。」「難しい。周りの子は終わってるのに、私はずっと逡巡してて、TAさんに何回か質問してしまった。もっとうまくできるようになりたい。」「もうちょっとゆっくり説明してくれると助かります。」などの意見があったが、2016年度は、「難しそうに感じていた分、意外と楽しめたので、次回以降も積極的に参加していきたいです。」「今日の進捗度合いがよかったです。」など、殆どの学習者は、問題なく課題に取り組んでいることが分かる。

## 5.2 今後の授業での活用案

本システムの基本的な利用手順については、有効に機能していることが確認できたため、2016年度の授業では、本システムの機能を活かした次のような活動を行う予定である。

- ・学習者間相互評価：学習者の進捗に応じ、数回、学習者間でプログラムの確認、評価を行う予定である。まずは、簡単な基礎課題において変数名やコメントを自由に付けさせて提出させ、読みやすいプログラムか否かの評価を相互に行う予定である。最終的には、応用課題等、少し複雑なプログラムについての評価を行うことができると考えている。同じ設問に対し、複数の解き方があることを具体的に理解することもできるものと考えている。
- ・学習者同士の QA：学習者が気軽に相談しあえる相談カテゴリを「ギャラリー」タブに設け、質問とそのコメントを学習者同士で行うことができる場の提供を想定している。

プログラミングスキルとして、プログラムを読む能力、変数等の値の変化を追う能力、書く能力に着目した研究があるが、本研究では、まずは授業の導入を通して、少しプログラムを書くことに慣れてきた初学者に対し、次のステップとして、互いのプログラムを読むこと求め、プログラミングの学習構成を総合的に検討していきたいと考える。

## 6 まとめ

本研究では、初学者の授業利用という前提を踏まえ、教師のプログラムや資料を閲覧し、自らがプログラムを書き、実行する、という一連の手順を、機能毎で三分割された画面を一画面に統合して表示する学習環境として開発した。実際の授業で試行したところ、学習者の満足度は高く、有効に機能していることが確認された。

本学習環境は、Moodle プラグインとして開発しており、Moodle コース内で利用することができる。本システムは、学習者が公開したプログラムを他者が簡易的に評価する機能を有する。また、教師を含めた他者のプログラムを自身のライブラリとして取り込むことも可能である。

これらの機能を用い、今後、学習者間でのプログラムの相互評価を通じた学びを展開する予定である。

## 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] 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10月21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/siryu1.pdf>, (2016年10月21日閲覧).
- [3] 教育課程部会情報ワーキンググループ, 平成 27 年 10 月 22 日資料 8 情報教育に関連する資料, 16 ページ, [http://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo3/059/siryu/\\_icsFiles/fieldfile/2015/11/11/1363276\\_08\\_1.pdf](http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/059/siryu/_icsFiles/fieldfile/2015/11/11/1363276_08_1.pdf), (2016年10月21日閲覧).
- [4] 布施泉, 岡部成玄, 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, PC カンファレンス 2016, pp.257-pp.260, 2016, <http://gakkai.univcoop.or.jp/pcc/2016/papers/pdf/pcc066.pdf>, (2016年10月21日閲覧).
- [5] 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 回全国大会講演論文集, pp.89-pp.90, 2016.
- [6] 加藤優哉, 松沢芳昭, 酒井三四郎, 独立同期モデルによる初学者向け協調プログラミング支援システムの開発, 情報教育シンポジウム 2014, pp.27-pp.34, 2014.
- [7] 関谷貴之, 山口和紀, 山本三雄, 初学者によるプログラムトレーニングにおける誤答に関する分析, 情報教育シンポジウム 2012, pp.113-pp.120, 2012.
- [8] 長慎也, 保福やよい, 西田知博, 兼宗進, De-gapper - プログラミング初心者の段階的な理解を支援するツール, 情報処理学会論文誌, Vol.55, No.1, pp.1-12, 2014.
- [9] elFinder, <http://studio-42.github.io/elFinder/>, (2016年10月21日閲覧).
- [10] Ace, <https://ace.c9.io/>, (2016年10月21日閲覧).
- [11] [http://nucla-hokkaido.jp/post\\_4.html](http://nucla-hokkaido.jp/post_4.html), (2016年10月21日閲覧).