Vol.2015-GN-94 No.5
2015/3/12

入門者向けプログラム学習用言語 EasyPL と Web 学習システムの提案

猪狩知也[†1]　速水治夫[†2]

近年，インターネットやコンピュータの著しい発達によりアプリ開発や Web ページ作成などを行う人が増えている．しかし，興味を持ってプログラミングの学習を初めても挫折してしまう入門者が多い問題が挙げられる．これは，プログラミングを始めるまでに開発環境の準備に相当な時間が必要であることや，バグが発生した際に入門者が一人でエラー文を読み，バグの原因を探しだし修正することが困難であることなど，様々な問題が存在する．問題を解決する着眼点として，プログラミングの敷居を下げて挫折するのを阻止する必要があると考えた．プログラミングの敷居を下げるために，学習用のプログラミング言語 EasyPL と Web ブラウザのみで言語の学習を行える Web システムを実装し，評価を行った．

Programming language for learning and Web learning system for beginners

KAZUYA IGARI[†1]　HAYAMI HARUO[†2]

In recent years, a growing number of people to do such as application development and Web page created by the remarkable development of the Internet and computer. However, beginners also the first time the programming of learning lead to frustration is problematic, and the like. And it is necessary a lot of time in constructing a development environment, alone when a bug occurs to read the error statement, such as it is difficult to fix out looking for the cause of the bug, various problems exist. As Viewpoints to solve the problem, it was considered to lower the programming of the threshold, it is necessary to prevent you to frustration. To reduce the programming threshold, implement a Web system that allows the programming language and a Web browser EasyPL only language learning for learning and evaluated.

1. はじめに

近年，インターネットやコンピュータの著しい発達によりプログラミングが一般的になっている．例として，プログラマーや SE といった仕事としてプログラミングをする人が増加したことや，iPhone，Android といったスマートフォンのアプリケーションを作りマーケットに公開するなど，様々なプログラミングを行う人が増えている．
また，RubyWarrior[1]や paiza.IO[2]，AtCoder[3]といったプログラミング学習サービスも登場している．これらの事からプログラミング学習の必要性は高まっている．

プログラミング入門者は，学びたい言語にもよるが基本的には学習用の書籍や Web 上の情報を元に独学か，大学や専門学校の講義，会社の研修などで学び始めることが多い．講義や研修でプログラミングの学習を始める場合は必ずやらなければならないが，独学の場合は学習者の学習意欲次第で学習量が変動する．

その為，学習意欲が高いプログラミング入門者はバグやエラーが発生した際に解決方法を Web 上で探すことや，試行錯誤を繰り返して解決し学習を続けることができる．しかし，興味を持ってプログラミングの学習を始めた学習意欲が高くないプログラミング入門者はバグやエラーが発生した際に解決できず，先に進めずプログラミング学習を辞める事が起こりうる．

そこで本研究では，プログラミング入門者用のプログラミング言語 EasyPL と Web 学習システムを提案する．本研究の目的は，プログラミング学習の際に発生する手間や無駄を出来る限り省く事により，プログラミング入門者の学習意欲を低下させないことである．

2. 研究対象の現状

2.1 プログラミング学習

現在のプログラミング学習は，プログラミング学習用の書籍や Web 上のプログラミング学習サイトを使い，プログラムの記述方法を学ぶ．プログラムの記述方法を学んだ後に，学習用の書籍や Web サイトに記述されているサンプルプログラムを写すか，正しく学べたか確認するための問題に挑戦しプログラムを記述して学習する．

書籍や Web 上の学習サイトでは，記述したプログラムの正答が乗っているが，学習者が書いたプログラムが正しく動くかの判定に使うテストケースが少なく，記載されているケースでは正しく動くが他のケースでは動かない場合が存在する．

†1 神奈川工科大学大学院
Graduate School of Kanagawa Institute of Technology

†2 神奈川工科大学メディア学科
Kanagawa Institute of Technology

2.2 プログラミング学習サービス

プログラミング学習サービスとは，Web 上でプログラミングを行えるサービスの事である．近年では，問題がクエストとして与えられ，プログラミング学習者はクエストをクリアできるように Ruby のコードを書き足して実行する Ruby Warrior や，学習用の書籍や Web サイトに書いてあるプログラムを実行して学ぶことができ，外部 API への接続やスクレイピングなど，学習以外にも使用することができる paiza.IO，競技プログラミング形式でプログラミングの問題に挑戦できる Web サービスである AtCoder などが挙げられる．

3. 問題点

プログラミング学習時に起こりうる問題点として以下の 4 点の問題点を挙げる．

3.1 環境構築

プログラミング学習を始める際には，エディタやコンパイラ，実行環境といった事前準備が必要である．これらをインストールし，設定するのに多大な時間が必要となる．

プログラミング入門者にとっては正しい手順であっても時間が掛かり，手順を間違えた場合は更に時間が掛かり，入門者のやる気を削ぐといった問題が挙げられる．

3.2 プログラミング以外の学習

プログラミング学習ではプログラムの記述だけでなく，コンピュータの知識やアルゴリズム，コンパイル時の仕組み, ポインタなど幅広い知識を勉強することが求められる．

しかし，プログラミングをやりたいプログラミング入門者に対して，最初から幅広い知識を学ばせるのは学習意欲を削ぐ原因になりうる．幅広い知識を学習するのは，プログラミング学習後，もっと深く学びたい入門者に対して学習を始めても遅くなく，入門者にプログラミング以外の学習のみに絞って学習させられない問題が挙げられる．

3.3 バグの修正

プログラミングでは，入門者から熟練者まで少なからずプログラムにバグを発生させる．プログラミングに慣れている者であれば，エラーの原因を過去の経験やエラー文から読み取る，デバッグをするなどして原因を特定する事ができる．例として，Java 言語で NullPointerException エラーが発生した際にはエラーが発生している行周辺で変数の初期化を忘れていないかを調べる．しかし，プログラミング入門者にとってエラー文を読んで原因を探すという行為は非常に難しい．そのエラー分が何を示しており，何処の箇所が悪いのかを判断するのに時間が掛かり，学習が進まず学習意欲をそがれる問題がある．

また，書いたプログラムがコンパイルエラーや実行時にエラーが発生せず意図しない結果を実行する場合がある．エラーは発生しなかったが実装を間違えた場合である．

実装を間違えた場合，エラー文が表示されないので何処にバグがあるのかをプログラミング入門者が自分で探さなければならない．このバグを探す行為も，プログラミング入門者には難しく，学習意欲を削ぐ問題が挙げられる．

3.4 プログラムの正当性

プログラムを書き終えたら，正しくプログラムが書けたかを確認するために，入力に対して出力が正しく返ってくるかのテストを行う必要がある．書籍や Web 上の学習サイトでは，テストケースが少なく，記載されているテストケースでは正しく動くが他のケースでは動かない場合が存在する．プログラミングに慣れている人であれば，必要なテストケースを自分で考えてテストし，プログラムを確認することができる．しかし，プログラミング入門者は必要なテストケースが分からず，自分でテストをすることができない．その為，書いたプログラムの正当性を保証できない問題がある．

4. 解決策

3 章で挙げた問題点を解決するために，プログラミングの敷居を下げて学習意欲を低下させず，挫折しないようにする必要があると考え，プログラミング入門者用の言語 EasyPL と Web システムを提案する．

自作言語 EasyPL では，分かりやすい言語の記法とエラー文を提供する．プログラミングを始める際に覚える構文やエラー文を少なくすることで, 3.2 節で挙げたプログラム以外の学習と 3.3 節で挙げたバグの修正のエラー文が分からない問題を解決する．

Web システムは，自作言語 EasyPL をブラウザ上で学習出来るように学習用の問題解答機能，Web エディタ機能，Web コンパイル機能，Web デバッグ機能を実装した．

学習用の問題解答機能では，学習用の問題が用意されておりプログラミング入門者が書いたプログラムを Web システムに送信することによって，あらかじめサーバ上に用意されていたテストケースを確認し，正しくプログラムが書けているかを確認する．この機能によって 3.4 節で挙げたプログラムの正当性の問題を解決する．

Web エディタ機能では，ブラウザ上でプログラミングをする際にテキストエディタや統合開発環境のようなエディタを使えるようにし，プログラミング入門者でもプログラムを書きやすいように行番号の表示，予約語の色が変わるなどの機能を実装した．

Web コンパイル機能では，ブラウザ上でプログラミングしたプログラムを Web 上のエディタから送信し, サーバ上で受信したプログラムをコンパイルする．コンパイルに成功したら，リダイレクトを利用して入出力を行ない，プログラムのテストを行う機能である．

Web エディタ機能と Web コンパイル機能を実装するこ

とにより，プログラミングを始めるまでに必要な環境構築をなくし，3.1 節で挙げた環境構築の問題を解決する．

Web デバッグ機能では，プログラムを 1 ステップごとに実行した際の変数の中身の挙動を見ることができる機能である．コンパイルエラーは発生しないが，プログラムの挙動が間違っている際に何処に原因があるのかを探すデバッグを効率よく行えるようにする．

この機能により，エラー文が出ない実装ミスのバグが発生した際に 1 ステップずつ見ていくことで考えていることと間違っている箇所を発見しやすくし, 3.3 節のバグの修正の問題を解決する．また，プログラムのステップ毎の挙動を可視化することにより，プログラムの挙動を確認することができる．その為，基本的な操作に分割しそれらの順序を意識するアルゴリズム的思考力[4]を鍛えることもできる．これらにより，プログラミング入門者がプログラミングを学ぶ際の問題を解決し，プログラミング学習を続けられるようにする．

5. 試作システム

5.1 システムの概要

試作システムはデータベース部，コンパイラ部，Web システム部の 3 つで構成されている．

データベース部では，ユーザの情報と問題の情報，提出したプログラムの実行結果の管理を行う．

コンパイラ部では，Web システム上で提出されたプログラムに対して字句解析, 構文解析を行い命令に分割を行い，プログラムを擬似的に処理し結果を保存する．

Web システム部では，ログイン，新規登録，問題表示，プログラム提出, 結果確認, デバッグを行うことが出来る．

5.2 画面遷移図

Web システム部の画面遷移図を図 1 に示す．

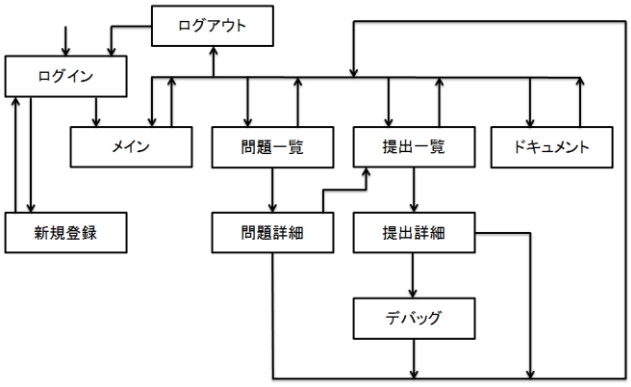

図 1 画面遷移図
Figure 1 Screen transition diagram

5.3 EasyPL コンパイラ

EasyPL 言語で書かれたプログラムは，EasyPL コンパイラによって命令に変換し，実行処理を行う．EasyPL コンパイラでは，プログラムの字句解析を行ない，int や for といった予約語や変数名など，意味のある文字列に分割する．字句解析を行った後に，意味が合うように正しく繋がっているかの構文解析を行う．構文解析では，擬似アセンブラ命令を生成し命令とその命令が対応しているプログラムの行番号を保存することによって実行可能なプログラムを生成する．

実行時には，命令を順番に処理していく．順番に処理していく際に，行番号に変更が発生したらメモリ上にある変数の状態一覧をテキストファイルとして出力する．この機能により, 出力されたデータを基に Web デバッグを実装している．

6. 評価と考察

6.1 言語の評価

EasyPL を学んだ後に, 実験協力者である研究室の学部生 6 人を A グループと B グループの 2 グループに分けて問題を 2 問解いて貰い，解答までに掛かった時間と提出した回数を記録する．

問題は不正解となるプログラムとテストケースが与えられており，テストケースが正しい結果を返すようにプログラムを修正する方式で行う．

A グループは 1 問目に EasyPL，2 問目には C 言語を使用し，B グループは 1 問目に C 言語，2 問目に EasyPL を使用して評価を行った．

6.2 システムの評価

また，評価後に試作システムが学習環境として適しているかのアンケートを行った．アンケートは下記の 4 項目と試作システムに対しての自由記述で構成されている．

I. EasyPL の言語仕様は分かりやすかったか
II. 環境構築の手間は減ったか
III. デバッグ機能は使いやすかったか
IV. 学習環境として試作システムは使いやすかったか

6.3 評価結果

グループ A の 1 問目と 2 問目を解くまでに掛かった時間と提出した回数を表 1 に，グループ B の 1 問目と 2 問目を得までに掛かった時間と提出した回数を表 2 に示す．

表 1 グループ A の結果
Table 1 Result of Group A

	EasyPL		C 言語	
名前	問題 1 正解時刻	問題 1 提出回数	問題 2 正解時刻	問題 2 提出回数
A1	11:15	3	29:03	2
A2	24:49	3	34:48	1
A3	10:57	3	13:33	2
平均	15:42	3	25:48	1.6666

表 2 グループ B の結果
Table 2 Result of Group B

	C 言語		EasyPL	
名前	問題 1 正解時刻	問題 1 提出回数	問題 2 正解時刻	問題 2 提出回数
B1	21:25	5	30:48	2
B2	40:43	3	45:02	2
B3	8:44	3	6:00	1
平均	23:37	3.6666	27:16	1.6666

言語の評価後に行ったアンケートによるシステムの評価を表 3 に示す.

表 3 アンケート結果
Table 3 Questionnaire result

	A1	A2	A3	B1	B2	B3	平均
I	4	4	4	5	4	4	4.16
II	5	5	5	5	5	5	5
III	4	3	3	2	4	4	3.33
IV	5	4	5	4	4	4	4.33

自由記述では，下記の意見を得ることができた.

- 全ステップ数の表示とステップの指定ができると更に使いやすくなる
- デバッグ機能に 1 ステップ毎とは別に指定したステップへ飛ぶ機能があるとより便利
- エラーが発生しているステップがわかると良い
- 提出した後に提出結果を wait と表示しないようにして結果を直接表示して欲しい
- 1 ステップ進むためにスクロールさせるのが手間なので, ボタンを上に配置し変数一覧をすぐに確認できるようにして欲しい
- 開発環境の構築をするのが苦手だったので助かった
- エディタのテーマ変更が良かった
- ステップを進める行を指定したい
- 現在のステップの行をソースコード側にもわかるように反映して欲しい
- 初心者は使いやすいと思う
- ステップ数が 1 つしか進めないのは不便
- デバッグ画面で実行結果と正答を確認できると良い
- デバッグ画面からソースコードを修正して再提出できると良い

6.4 考察

言語の評価では，1 問目が EasyPL の方が 8 分程速く，2 問目が C 言語の方が 1 分半程速い結果となった．提出回数には大幅な違いは見られず，どちらかの言語が優れているといった結果を得ることが出来なかった.

システムの評価のアンケートでは，設問 C 以外では平均 4 以上となる評価を得られ，プログラミング入門者に使いやすい言語と Web システムが提供できたのではないかと考えられる.

しかし, 設問 C では 3.33 と他の評価と比べ良くない結果となった.

原因として，自由記述欄にデバッグ機能の改善について多く書かれている．デバッグ機能の画面構成やユーザビリティを追求する必要があるが，これらの問題を解決し使いやすいデバッガを用意できれば 3 章で挙げたプログラミング入門者にとっての問題点を解決できたと考えられる.

7. おわりに

本研究では, プログラミング入門者の為の言語 EasyPL と学習用の Web システムを提案した．評価実験の結果，試作したプログラミング言語 EasyPL とデバッグを除く学習用の Web システムは高評価を得られた結果となった. アンケートの自由記述欄に多くの評価者がデバッグ機能の改善についての意見が寄せられた.

試作システムはデバッグ機能を改善することにより，よりプログラミング入門者が学びやすく，バグ修正に時間をかけずに良くなる環境が作れると考えられるので，デバッグ機能の改善を行っていきたい.

参考文献

1) RubyWarrior https://www．bloc．io/ruby-warrior/
2) paiza．IO https://paiza．io/
3) AtCoder http://atcoder．jp/
4) アルゴリズム的思考法の教育(情報処理学会研究報告．コンピュータと教育研究会報告) p．58