

プログラミング入門者向けのオンラインジャッジシステム によるプログラミング学習支援

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プログラマーといった職業やスマートフォン用のアプリケーション, ゲーム制作, Web ページ作成などを行う人が増えている. また, Code.org 等の非営利団体がプログラミング学習を必修科目にする活動を行っており, プログラミング学習の必要性が高まっている. しかし一方で, 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統合開発環境やコンパイラのダウンロード, インストール, 設定などの下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, プログラミング入門者向けのオンラインジャッジシステムを提案する.

Learning support by online judge system for programming beginners

KAZUYA IGARI^{†1} HARUO HAYAMI^{†2}

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きている. 例えば, 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ことができるようになったり, コンシューマや携帯機, パソコン用のゲームを作成したり, 会社や個人の Web ページ, ブログ作成などを行う人が増えている. その他にも, 機械学習やデータマイニングなど, 様々な分野でプログラミングが活躍する機会が増加している.

また, Code.org[1]などの非営利団体がプログラミング学習を必修科目にする活動を行っていることや, RubyWarrior[2]や Codecademy[3]など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非常に高まっていると考えられる.

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で学ぶか, 大学や専門学校の講義で学ぶことが多い. 講義である場合は単位を取る為に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と考えられる.

その為, 学習意欲で満ち溢れている入門者はバグやエラーが発生しても解決できるまで Web 上で解決方法を探ることや, 試行錯誤を繰り返して解決することができる. そ

の為, 学習意欲がある入門者はどのような環境であったとしてもやり続ける人が多いが, 興味を持ってプログラミングの学習を初めた入門者は単純な要因ですら学習意欲を削がれて挫折してしまうこと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統合開発環境やコンパイラのダウンロード, インストール, 設定などの下準備に相当な時間が必要であるといった問題や, 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問題など, 様々な問題が存在する.

これらの問題が発生することにより, プログラミングを学んでみようと思った入門者の学習意欲を奪い取ってしまい, 上手く行かないからプログラミングを辞める, 分からないからプログラミングを辞めるといった事態が起こりうる.

興味を持ってプログラミングを始めた入門者や, 授業などでプログラミングを始めた入門者が様々な問題により辞めてしまう現状を解決することにより, プログラミングを辞めてしまう人を減らせると考えた.

そこで本研究では, プログラミング入門者向けのオンラインジャッジシステムによるプログラミング教育を提案する. 本研究の目的は, プログラミングの学習意欲を削ぐ問題を解決することにより, 効率よくプログラミングの学習を行うことができるようにすることである.

提案するオンラインジャッジシステムでは, コンパイラの違いや OS の違いを発生させないようにし, メモ帳 1 つ

^{†1} 神奈川工科大学大学院 工学研究科 情報工学専攻

^{†2} 神奈川工科大学 情報メディア学科

から始められる Web コンパイル・実行機能, エラーを解りやすく表示する為の構文解析機能, プログラムを 1 行毎に実行してメモリ間の挙動や標準入出力を可視化してプログラムの挙動を解りやすくする Web デバッグ機能を備えている,

以下 2 章で研究対象の現状, 3 章で関連システム, 4 章で問題点, 5 章で解決策, 6 章で試作システムの概要, 7 章で終わりに を述べる.

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, 主にプログラミング学習用の書籍を使っての学習かインターネット上に公開されているプログラミング学習用の Web サイトで学習することが多い. どちらの場合においても, プログラミング入門者が書籍上のソースコードを一字一句間違えずプログラムを記述し, プログラムが正しく動作することを前提として書かれていることが多い.

2.2 オンラインジャッジシステム

オンラインジャッジシステムとは, 大量の問題が与えられてその中から問題を 1 つ選び, 選んだ問題の問題文や制約を確認し, プログラムを実装して提出するとソースコードの正誤を判定してくれる Web システムである.

オンラインジャッジシステムで使用される問題は, 入門者にとっては非常に難しい問題ばかりである. 条件分岐やループ処理だけで解ける問題であったとしても, アルゴリズムを考察する思考力が足りないと解けないからである.

3. 関連システム

3.1 Code.org

Code.org では, 全ての学校の全ての学生がコンピュータ・サイエンスを学ぶ機会を与えられるべきだという理念の元活動しているプログラミング学習サービス. ソースコードを書かずにブロック状の物体を組み合わせて鳥を動かすプログラムを記述し学ぶことができる.

3.2 RubyWarrior

RubyWarrior では, Ruby のソースコードを記述してキャラクターを動かすことができる. 遊びながら Ruby の基礎を学ぶことができるプログラミング学習ゲームである.

3.3 Codecademy

Codecademy では, ブラウザ上でプログラムを記述し, HTML, Javascript などの Web 系のプログラミング学習を行うことができる. 問題によるが, Web 系のプログラミング

言語学習なので, プログラムを書いている最中に結果を見ながらコーディングすることができる.

3.4 AtCoder

AtCoder では, 競技プログラミングの問題を解く事が出来るオンラインジャッジシステムである. 最近はプログラミング入門者向けの AtCoder Beginner Contest も行っている.

4. 問題点と解決策

4.1 始めるまでに必要な準備の時間が長く難しい

プログラミング学習をする際には, エディタやコンパイラ, 統合開発環境など様々な下準備が必要である. これらを準備して正しく動作させるのに多大な時間と場合によってはエラーが発生し進まない場合がある. 例として Windows のコンピュータにて Java 言語での学習を始めようとする. 例としてエディタとコンパイラを使う例を挙げる. まず初めに, 好みのテキストエディタと最新の JDK をダウンロードし, インストールする. その後, JDK のパスを環境変数に追加してプログラムを書き始める. プログラムが完成したら, コマンドプロンプトを起動してディレクトリを移動し, javac コマンドでコンパイルし java コマンドで実行する. 初めてプログラミングを行うユーザがこの手順を間違えずに行えたとしても相当な時間が必要であるし, 何処かで手順を間違えた場合は更に時間が掛かる. 全て上手く行ってプログラムを実行した結果が HelloWorld しか出力されず, 時間を使って頑張ってもこの程度しかできないなら辞めようと思って挫折してしまうプログラミング入門者が存在する.

4.2 バグが発生した際に自己解決するのが困難

プログラミングをしていると, 入門者から熟練者まで, 少なからず何かしらのバグを発生させたことがあると思われる. ある程度プログラミングに慣れている人なら, エラーの原因をエラー文やエラーコードから読み取り, 何処でバグが発生しているのかを特定することができる. 例えば Java 言語で NullPointerException エラーが発生した場合には, 慣れているユーザならエラーが発生した行に関わっている変数の初期化忘れがないかをまず調べるであろう. しかし, 入門者にとってはエラー文を読むという行為は非常に難しい. そのエラー文が何を示しており, 何処が悪いかを判断するのに時間が掛かり, プログラミングは敷居が高いと挫折してしまうプログラミング入門者が存在する.

また, コンパイルエラーや実行時エラーが発生せず, プログラミング入門者が意図しない結果を実行する場合がある. これは, エラーは発生しなかったが実装を間違えてしまった場合である. この場合では, エラー文も表示されないで何処にバグがあるのかを自分自身で探さなければな

らない。この探す行為はプログラミング入門者には非常に難しい問題である。

4.3 プログラムの正当性を保証することができない

4.2 のバグが発生した際に自己解決するのが困難で挙げたように、プログラム自体にバグがあるかないかを保証することができない問題が挙げられる。例として、ソートのプログラムを実装した際に正しく書けたかをテストする例を挙げる。プログラミングに慣れている人であれば、バグが存在しないかを確認する為に必要なテスト項目を自分で考えてテストし、確認することができる。一方で、プログラミング入門者は必要なテスト項目を自分で考えることは非常に困難であり、どのようなテストをすればいいかわからない。また、書籍や Web サイトにもテスト項目が記述されている場合があるが、テスト項目数が非常に少なく、書いたプログラムの正当性を保証することが出来ない可能性が高いといった問題がある。

4.4 必要以上の学習を入門者に求めている

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やポインタなど幅広い知識を勉強することが求められている。しかし、単純にプログラミングを学んでみたいと思っているプログラミング入門者に対して、幅広い知識を学ばせることを強要するのは学習意欲を削がれる原因になりうると考えられる。幅広い知識を学習させるのは、プログラミングを学習してもっと深く学びたいと思ってから幅広い知識を学習し始めても遅くなく、入門レベルの状態であるべきでないことはやらせないようにした方がいい問題がある。

5. 解決策

5.1 解決策

4 章で挙げた問題点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学習意欲を低下させず、挫折を防止させる必要があると考えた。

そこで本研究では、プログラミングの敷居を下げるための 3 つの機能を持つプログラミング入門者向けのオンラインジャッジシステムによるプログラミング教育を提案する。3 つの機能により、効率よくプログラミングの学習をすることが出来るようにすることである。

5.2 Web 上でプログラムをコンパイルし実行

1 つ目の機能として、Web 上のフォームからプログラムを送信し、送信されたデータをサーバが受け取り、サーバ上でプログラムをコンパイルする。その後、生成された実行ファイルかクラスファイルを元にプログラムを実行し、

リダイレクトを利用して標準入出力を行う。その結果を返して Web 上で表示することにより、Web 上からプログラムをコンパイルし実行する機能である。

この機能により、プログラミング入門者は最低限必要な環境として、Windows に付属しているメモ帳と Internet Explorer を使うことでプログラミングを始めることが出来る。プログラミング入門者はソースコードを書くだけでコンパイルして実行することが出来る為、すぐに始められるしコンパイル時のコマンドを覚えなくてもよい。

また、コンパイルエラーが発生した際には通常では分かりにくいエラー文より分かりやすいメッセージで何処に不具合があるかを指摘してくれる機能も付属している。

5.3 変数間の挙動と入出力を可視化した GUI デバッグ

2 つ目の機能として、変数間の挙動と入出力を可視化した GUI デバッグ機能である。この機能では、プログラムを 1 ステップ毎に実行し変数の挙動をグラフィカルに見ることが出来る。コンパイルエラーは発生しないが、プログラムの挙動が正しくない際に何処に原因があるのかを探すデバッグ作業を行いやすくする。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実装した本人が考えていることと間違っている箇所を発見しやすくなると考えた。

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⁾を鍛えることが出来る。

5.4 複数のテスト項目を確認して正当性を保証する

3 つ目の機能として、問題に対して複数のテスト項目を確認し、全て正しかった際にプログラミング入門者が書いたプログラムの正当性を保証する機能である。この機能では、プログラミング入門者が試作システムにプログラムを投稿し、コンパイルに成功した際に生成された実行ファイルに対してあらかじめ用意されている大量のテスト項目を多数試し、正しい結果を返すかを確認する。全て正しい結果を返した際に、プログラムは与えられた制約内にて正しく動くことを試作システムが保障する事が出来る。

この機能により、プログラミング入門者がプログラムを書いた際に、4 章で挙げたコンパイルエラーや実行時エラーは出ないけど間違っている可能性がある問題に対し、大量のテスト項目を試して正しいかどうかを判定し、正しいプログラムを書けたことを保証することができる。

6. 試作システムの概要

6.1 システムの概要

試作システムは標準入出力のやり方から条件分岐やループ文などのプログラミングの入門部分を書く問題を問題として登録されたオンラインジャッジシステムを実装する。

また、5章の解決策で提示した3つの機能の実装を行い、学習意欲を低下させないようにする。

6.2 問題解答機能

プログラミングの問題を解いていき、学習することができる機能である。この機能では、実際にプログラムを書くことやデバッグをすることにより、プログラミング能力の向上を狙う。プログラミングのコンパイルや実行は全て試作システムで行うため、プログラムを書くテキストエディタと試作システムを使用する為の Web ブラウザさえ用意すればすぐに始めることができる。また、プログラムを提出した際には問題毎に登録されている大量のテスト項目を実行する為、正答だった場合に記述したプログラムの正当性を保証する事ができる。

6.3 Web デバッグ機能

Web デバッグ機能では、実行するソースコードと与える標準入力を選択することにより、Web 上でデバッグすることができる。プログラムを1行1行進めてメモリの挙動と標準入出力を確認することができる。この機能を利用することにより、プログラム実行時のエラーの際はどこでプログラムが停止しているのかを把握することができ、原因を調べやすくなる。実装ミスによるバグの検知も1行1行進めてメモリ間の挙動を確認や、意図しない挙動を見つけることにより解決しやすくなると考えられる。

また、問題の解答の挙動を確認する機能も実装予定である。解答の挙動を確認することにより、プログラムをどのように構築する参考例を、ソースコードを見せずに伝えることができる。

図1に Web デバッグ機能の画面図を示す。

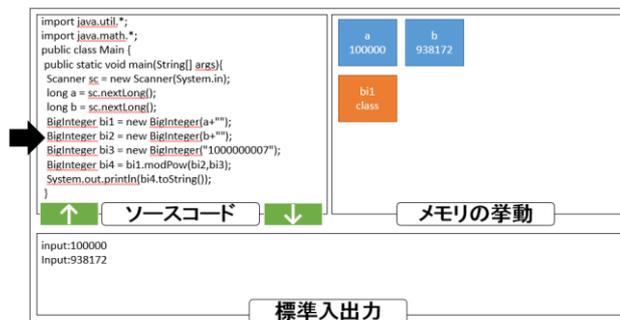


図1 Web デバッグ機能画面図

7. 終わりに

本研究では、プログラミング入門者が学習中に挫折しないようなプログラミング学習の為のオンラインジャッジシステムを提案した。提案システムは Web システムとして実装し、いつでも、どこでも、誰でも使えるように実装を行う予定である。

参考文献

- 1) Code.org 入手先<<http://code.org/>> (2014,05,16)
- 2) RubyWarrior 入手先<<https://www.bloc.io/ruby-warrior/>> (2014,05,16)
- 3) Codecademy 入手先<<http://www.codecademy.com/ja/>> (2014,05,16)
- 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)原稿執筆案内, 入手先<<http://ci.nii.ac.jp/naid/1100066683961/>> (2014.03.25).